Date of acceptance Grade

Instructor

Mobile Business Processes: Challenges, Opportunities and Effects

Aikeremu Tiemuer

Helsinki June 12, 2017

UNIVERSITY OF HELSINKI
Department of Computer Science

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Computer Science

Tekijä — Författare — Author

Aikeremu Tiemuer

Työn nimi — Arbetets titel — Title

Mobile Business Processes: Challenges, Opportunities and Effects

Oppiaine — Läroämne — Subject

Computer Science

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
	June 12, 2017	55 pages + 0 appendices

Tiivistelmä — Referat — Abstract

Smartphones and tablets have become an important part of people's daily life and people have started to use them for work purposes as well. Mobile computing has location and time flexibility advantages that enterprise can utilize with care. The world of mobile technologies is fragmented. It can be a complex task for a non-mobile software company to decide what kind of smartphone application they want for their customers. People have different expectations on mobile applications and their usage habits differ. Mobile applications are more prone to disturbance because of their nature and again still performing critical tasks. They cannot be treated same as stationary computers in terms of business process support.

This thesis presents an analysis of effects and opportunities brought by mobile devices to enterprises from small to large scale based on the literature survey. The thesis helps enterprises to understand what kind of approach to mobile support best suits their needs and how to handle the mobile-specific challenges in their IT systems.

ACM Computing Classification System (CCS):

Applied computing –> Enterprise Computing –> **Business-IT alignment**,

Information systems –> Information systems applications –> *Mobile information processing systems*

Avainsanat — Nyckelord — Keywords

Mobile, Business process, Software engineering, Mobile services, Workflow

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Contents

1 Introduction

Mobile computing is becoming more significant in people's daily life and is replacing traditional desktop computers both in numbers and usage. According to Gartner (2013), almost 1.9 billion mobile phones and tablets were sold worldwide in 2012 and PC sales were only 341 million [RGG13]. Both mobile devices and broadband connections are widely available. People use online services more likely on a mobile device than PC nowadays, for example, over half of YouTube views in 2014 from mobile and mobile revenue on YouTube is increased by 100% in 2014 [You14].

Modern mobile computing devices are special in the way that they are light, can always be carried by their owners, always kept on and connected to the internet. Furthermore, they are packed with sensors such as GPS and camera. This increases the opportunities for traditional and new business applications for optimization of their involvement and current business models.

As mobile devices and user habits are different from desktop computers, mobile computing is currently facing a set of special challenges. The world of mobile operating systems is fragmented. Developing mobile applications and supporting them on all major platforms has significant costs. The users tend to expect simpler user experience on mobile devices than on desktop computers. However, the use of such applications is prone to disturbance because people use them while walking, in functioning signal circumstances and with low battery condition. A simple failure of a mobile application may still lead to serious consequences [BBC15]. Therefore, development of more reliable mobile business applications are essential.

A business process is a sequence of tasks and interactions, which is performed together by independent companies or organizations. Each task run either by computers or by humans. Business process is defined in a computer language using modeling tools by designers. Forming rules to be executed by process engine and run on servers [JEA07]. Mobile Business Processes are the business processes that enable user interaction through mobile devices.

Mobilizing business processes, adding mobile applications and enabling mobile user access to existing business processes, can release information access from time and location constraints. Therefore, it provides smoother information flow that improves efficiency. Mobile workers, e.g., people who do not work in the office, can transmit data back to the command center and receive updates in real time. Working this way saves costs and time mainly because of saving time in traveling in order to retrieve information. This also improves quality and flexibility of the service. This increases efficiency in information flow in different B2B (business-to-business), B2E (business-to-employee) and B2C (business to customer) scenarios [FaT14] [BoP08].

Traditional business processes and execution facilities were created for stationary computers and such solutions assume participants to have enough computing power to deploy business process engines and reliable network connections. Unfortunately, this is not true for mobile devices. Mobile devices have limited computing power and unreliable network connections. Currently, there is not a single full-fledged business process engine designed to run on mobile devices.

Enterprises traditionally serve their customers through a stationary web interface which must adapt to this development regardless of their scale. In simpler use cases, such as for booking an appointment, developing mobile web interface would be enough. More complex cases require deeper engagement from mobile users which requires more complex facilities such as a mobile business process engine.

Regardless of the popularity and impact of mobile computing to our life, the practicalities of providing a mobile solution to customers can be hard to understand. Mobile device market is very dynamic and still not stable. New devices from different vendors are released almost too frequently and they all come with different features and software platform support. Enterprises outside the mobile software business have little capacity to follow the trends in mobile development technologies. Even those with an IT department traditionally have facilities to support stationary user base and usually do not have in house competence to develop mobile applications.

While mobile business applications have begun to evolve, enterprises have become interested in business cases that have really benefited from mobile support. Enterprises want to understand the effects of mobile computing to their business and challenges they will face when implementing mobile solutions. This thesis develops a roadmap of utilizing the new possibilities mobile devices can bring and using them to improve business processes for the enterprises. This thesis begins by analyzing the options available for enterprises in order to help them decide whether or not they should implement a mobile solution. Then, the thesis examines the architectural choices for implementing a mobile application client in the most suitable cost effective way. Thereafter, the thesis evaluates the current solutions in integrating mobile devices into distributed execution of business processes seamlessly and reliably, and recommends solutions for the needs of varying enterprises.

This thesis is structured as follows. Chapter 2 analyzes the benefits and effects of mobilizing business processes. As a result, the most beneficial scenarios of introducing mobile are identified as well as the challenges of supporting mobile devices. Chapter 3 evaluates the ways to integrate mobile devices into the execution of the business process. Chapter 4 compares methods of developing a mobile application as well as software reuse. Chapter 5 is devoted to the challenge of robust execution of mobile tasks. Chapter 6 investigates the integration of more advanced knowledge-intensive processes using mobile business process engine. Chapter 7 summarizes the whole roadmap discussed and concludes the thesis.

2 Enabling of Mobile Computing

In plain words, enterprises have been left far behind consumers when it comes to embracing mobile devices. People are more likely to purchase mobile devices for their private use rather than for work. Business is still passively accepting this change rather than actively welcoming it. It seems obvious that the most successful mobile applications so far have been developed for private entertainment and social needs of individuals rather than for business.

Mobile IT is a less formal name for pervasive or ubiquitous computing. The meaning of these terms has evolved throughout time. In the 1990s they meant more portable computing devices such as laptops. Nowadays, they refer to the following common characteristics of smartphones and tablets: i) intuitive user experience, ii) wide availability and connectivity, iii) contextual computing ability supported by different sensors [SBB13].

Well-developed mobile applications are very efficient and easy to use compared to desktop applications [FWC14]. People can accomplish the same task more efficiently with better user experience. As a result, people have started to use their private mobile devices for business whenever they can, and a new trend called *IT Consumerization* has begun.

2.1 Mobile consumerization

IT Consumerization means that companies allow employees to bring their own mobile device (BYOD) to work, as well as to use some consumer applications such as Skype and Dropbox for work purposes. In 2010, 37.6% of all business phones were purchased privately. In 2011, 78% of American IT decision-makers accepted the use of private devices and applications in the company environment. These applications are very intuitive and people are already familiar with them because they have used them often in their private lives. These applications have changed people's expectation from their corporate IT and people have started to expect the same level of user experience from their company applications [WeL12] [KoC14].

Consumerization affects both corporate hardware and software. Apple's iPad and Microsoft's Surface tablets are replacing notebook PCs. Some elevated consumer applications for business are started to emerge, such as Gmail for work, Skype for business, Dropbox for business. Social networks such as Yammer are used for improving internal communications.

Mobile consumerization has imposed challenges to traditional corporate IT [WeL12]. On the one hand, using consumer devices and applications for business purposes can really improve internal communications and efficiencies by active use and intuitive applications require no or little training. On the other hand, these applications have not been designed and implemented with business users in mind and cannot fully satisfy business requirements.

In terms of security, there is a gray area between business and private information if the same device and maybe even the same user account has been used for both purposes. Preventing accidental information leakage and managing the security of increased amount of information have become central challenges [WeL12].

Although storing corporate data on individual devices can save some costs in data storage, security has become difficult to manage. This is because service providers store consumer data on virtualized data centers [WeL12].

Regarding information system management, the use of smartphones and tablets for business requires companies to redesign their IT systems. Originally, corporate tools and business processes are developed for stationary computers and do not provide a mobile experience. Most of the current consumer applications that run on mobile devices and have been used for business were originally designed for private users. They either lack real business-level quality or business license support. Innovations in the mobile market are mostly offered by small startup companies. These applications tend to be changed or eventually bought out by large companies. Therefore support and long-term development are not guaranteed. All of these risks must be managed [WeL12].

IT consumerization has brought challenges of managing many different types of consumer devices by corporate IT, and usually, mobile devices lack central IT management support.

IT departments must find the right balance between the usability and the security of the corporate IT systems. The existing infrastructures built over decades do not satisfy the employees from the usability point of view. However, the external innovations from consumer market that are easy to use are not secure enough for business. If IT departments and CIOs are ruthless enough to reject all the innovations from the consumer market, they create a gap between IT and other departments, and the other departments might seek external solutions for their problems [SBB13]. The key to the problem for IT departments is actively accommodating to new trends and supporting mobile devices, adapting existing business processes and making them more mobile friendly and also secure.

Instead, consumerization should be seen as a chance of IT departments to grow. Their focus can be shifted from cost saving to creating value. They can run more projects with smaller budgets such as developing a mobile application and target value-adding to company's productivity instead of saving cost [KoC14].

One approach to managing security is to define a flexible boundary between the core services, which company operations are ensured, and the customer interface, where the innovations are facilitated [SBB13]. IT departments maintain the core services and processes, which are build over decades at the heart of the company and are also most valued. Meanwhile, they ensure data safety and allow flexible and intuitive access to the services and data by developing additional mobile applications.

2.2 Effects and opportunities of mobile on businesses

Introducing mobile into business processes may also lead to some organizational effects [PoH09] [SBB13] [NBL10] [LGA12]:

- Mobile devices enable workers to have more freedom to choose workplace. They do not need to visit the office in the morning in order to fetch their schedule updates. They can even stay at home or be somewhere else where they want to finish their tasks.

- Mobile devices improve remote support and control. Support can be provided remotely and in time by experts. The head office has better control of their workers as long as they receive real-time updates of status information.

- Mobile devices can reduce office roles such as call center jobs. Remote workers can synchronize information regardless of their location, so there will be fewer phone calls to the head office requesting information.

- Mobile devices lead to a closer integration of partners and end-users and self-service. Customers can finish some tasks previously handled by staff such as creating orders or check-ins. This trend has more potential as the capability of devices are increasing and technology advancements.

- Using mobile devices result in more data collection for business intelligence. Previously companies rely on interviews to understand their customers. With mobile computing and sensor technologies, they can collect more data about customers and their user experiences.

Further effects of mobile devices are improved communication quality by merely using mobile devices instead of traditional workstations [SBB13]. Tablets and phones are mostly regarded as consumer products and therefore they are the less formal devices to communicate. Using them for video calls with customers can eliminate psychological barriers and create a relaxed atmosphere and therefore generate sales.

Although employees first started to use mobile devices for business purposes as they found it convenient, it does not necessarily mean they will actively use a mobile business solution provided to them [BeB14]. Their motivation and interest of using mobile equipment do not have to match the interest of their companies altogether. Supporting mobile devices in businesses and services has great potential for enterprises, but it also requires significant changes in their current IT system, infrastructure. So, it is essential to understand the primary motivations of implementing mobile solutions for enterprises and their employees.

Employees typically prefer mobile business solutions that are easy to use. They may, for example, expect touch-based mobile user experience which is built on consistent interaction pattern with other mobile applications on their devices. They also want to improve their social image by holding a popular device and being able to handle their work on that device.

Meanwhile, enterprises can reach greater user base in a shorter development cycle by developing a mobile client. They want to guarantee their information security by enforcing strict security policy on mobile devices such as entering long security pins every time devices are unlocked, and this might conflict with the interest of users who seek easy usability.

Enterprises should aim the right level of security and usability [SBB13] as emphasizing too much on security might compromise intuitive usability. Using very intuitive authentication means such as use of fingerprints instead of long pin codes might have great potential in this area which can provide uncompromised security while also not losing intuitive user experience.

However, there exist mutual interests of both employees and enterprises, namely, workspace flexibility, information availability and process mobility. Workspace flexibility means that employees can work from anywhere. Information availability refers to a situation where employees do not need to carry around lots of document and can retrieve information using their mobile device when needed. Process mobility means that employees can complete actual work on their mobile devices. All this efficiency and flexibility naturally increase enterprises competence in the market [BeB14].

Studies show IT consumerization can cause extra stress in people's private and work life because of following reasons:

- Employees use mobile devices for both private and work life will be exposed to increased reachability [OKM13]. Because they can receive work emails in their private time, they might feel obliged to check and answer. In some workplaces, this might be misused and cut off employee's free time.
- Still some people might find mobile software are hard to use considering they never have used them and had no interest to learn [OKM13].
- Instant messages, push notifications from all different social media and messaging applications are constant distractions for work and decrease productivity [SLe09]. On the other hand, work is expecting faster responses on tasks.
- If there are too many ways of communicating at work which is called system redundancies, employees can feel the extra burden of keeping track all of them and move their attention from their real job.

Thus, even being able to work from anywhere is not entirely suitable for everyone. Enterprises need to implement right policies and give training to their employees To utilize mobile devices at work efficiently while avoiding side effects. They still need to carefully evaluate implementing mobile solutions to prevent extra cost and adverse side effects of IT consumerization.

2.3 Feasibility study process

Mobile devices have the advantage of enabling people to work in flexible time and location. They also provide benefits and new opportunities to businesses. However, there are some side effects and unresolved issues that need to be considered. Not all business scenarios benefit from introducing mobile support. For this reason use cases need to be evaluated very carefully case by case.

Since enterprises ultimate goal is to improve their business process using the latest mobile technologies, it is beneficial to get a holistic view before making any pure technology-driven decision [HGK14]. This kind of holistic understanding can be gained through feasibility study by using some analysis framework such as the one provided by Hoos et al. [HGK14]. That analysis framework for identifying value added business scenarios. It primarily includes three activities based on criteria list similar to above and ends in numeric result calculated based on values output from each step. This framework can be used in case when more rigid analysis result is needed for comparison use.

However, enterprises can conduct a feasibility study in following criteria in order to assess whether a mobile solution case would create a service that would be the right nature and fit into the enterprise' strategy. These criteria include company's reality, technology support and cost-benefit ratio, etc.

Whether the nature of business suitable for mobile service? If the enterprise's core business is mobile by nature such as in logistics or construction industry, mobile devices are their only optimal choice because stationary IT cannot support them. Sometimes core business is not mobile but some essential functionality is [CNa08], for example, ambulance service of hospitals. There are two important characteristics of such mobile business functions, it is either employees are moving and approaching customers or service is happening outside of business's main facility. These can be considered as the input when determining the type of the business case [GoP09].

Is response time crucial for the task? Mobile support can improve response time [SGe88], but if the handling time is not critical for the work, then there is no significant gain from mobilizing it [GoP09].

Is it information intensive industry? Mobilizing can improve data availability and volume. Information intensive industries [GRa91] can benefit more from mobile applications [GoP09].

Does IT team understand the business process? IT developers tend to have more focus on technical details rather than the business object. Better usability expectations can be fulfilled if they have more focus on business details.

Does top management support the idea? Mobile development projects are more likely to succeed if the projects get top management support. Because during the project implementation, it is necessary to make changes to company's business processes and that cannot be achieved merely by IT developers without management support [HCh93].

How mature the mobile technologies and how good the application design? This is for checking the technology supports are in place, sometimes business idea can be too futuristic and it could be actually not possible to achieve it with currently available mobile technologies. For example, indoor navigation [RHM04] has been a topic with great interest, but still not yet mature implementation widely deployed due to technology limitation of mobile devices. Therefore, it is important to ensure needed APIs are provided by the target platform, the mobile broadband connection is enough for data communication needs or Wi-Fi coverage is available in all the area that is going to be used. It is important application design is simple yet useful enough to fulfill the requirement, so users are willing to use it.

Are needed devices available or possible to purchase? Can touch-based mobile devices, that have relatively small screens and limited computing power and battery capacity, fulfill the need of application? If not can cloud and offload to servers help?

Can mobile solution meet security requirement? Security includes valuable information stored in the device in case device is lost or stolen.

Can sensor features on the mobile devices such as camera or location bring valuable improvement to business? For example, it is easier to record a scene by just taking a picture rather than describing it in the report, also using location sensor instead of entering an address can improve precision.

Does it worth the cost? However, this question cannot be directly answered, and the answer may not be obvious in some cases. It needs to be noted that the cost includes planning, development and maintaining costs. The value includes monetary benefits and indirect financial benefits which are harder to calculate such as benefits come after mobile application improved overall effectiveness of the organization.

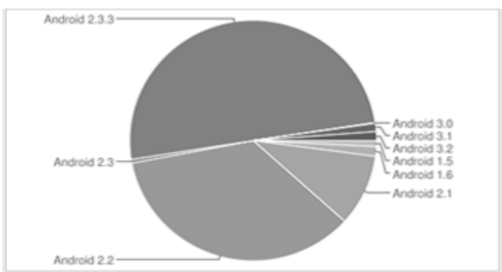
Overall understanding gained by answering each question listed above can be really useful input for final decision of whether investing for new mobile service creation project.

2.4 Challenges of implementing mobile solutions

Once after enterprises decide to support mobile devices, next step is project planning. It includes designing overall security architecture, determining mobile competencies either from in-house or external resources, choosing the mobile platform to develop on and support, estimating the time and budget for the project and make the optimum plan.

Unlike stationary computers, mobile devices have some special characteristics that require understanding while developing applications. This is especially the case with the more complex cases which demand longer sessions and collaboration from several service providers.

The mobile world is more fragmented than the PC world. There are much more variety of options and products exist. According to Gartner report [RGG13], worldwide device shipments by the operating system in 2012 Android, Windows and iOS/MacOS are top 3 operating systems accounted for about 23%, 16%, and 10% device sales. Besides, this figure includes tablets and PCs which applications usually not compatible with mobile even on the same operating system. Developing a native application and supporting on all the leading mobile platform is a rather costly project. Android world is very fragmented, and different device manufacturer uses their customized version of OS [HaP14] as shown in Figure 1. Therefore, to make sure the applications work on every Android device, extensive testing for compatibility is required. Windows and iOS seemingly do not have such problem, but they have different operating system version for tablets and phones which applications are not compatible.



Platform	Codename	API Level	Distribution
Android 1.5	Cupcake	3	0.8%
Android 1.6	Donut	4	1.3%
Android 2.1	Eclair	7	9.6%
Android 2.2	Froyo	8	35.3%
Android 2.3 - Android 2.3.2	Gingerbread	9	0.5%
Android 2.3.3 - Android 2.3.7		10	50.1%
Android 3.0	Honeycomb	11	0.1%
Android 3.1		12	1.1%
Android 3.2		13	1.2%

Figure 1: Android Fragmentation [HaP14]

Developing a mobile application for popular three mobile platforms and maintaining them is significant work with cost. The cost issue is the second top concern of enterprises who are going to implement mobile solutions. Therefore, cross platform development tools and techniques look very attractive from cost saving point of view.

Mobile applications are prone to disturbance and connection inclined to drop because users might switch application, e.g., to check new incoming messages or battery drain or because of the weak mobile signal. Making robust mobile business applications that work in such situation is challenging.

For example, assuming a user left his office with a particular keyword in his mind, and he tried to search the keyword online while walking on the street and encountered a specific product. Suddenly he saw an underground metro entrance and went down. Network connection on his is disturbed because of the cellular connection change from a base station to an underground antenna. He did not notice the change and continued shopping the product. While he is at last payment step, he received a push notification from messaging application and instantly tapped on it. Consequently, the application he was using for shopping went to the background. Because of battery saving policy on that smartphone, background applications will not be allowed to have active network connections, and that application went into the sleeping mode. After he had switched back to shopping application, he expects the application should recover the same status as he left. When all this just back to normal, smartphone automatically turned off because of the battery run out. Now the earliest chance he can charge his smartphone would be only after an hour later when he gets home. Does he even remember to continue shopping when he gets home and should application send a reminder? In case he wants, can he continue from the step where he was interrupted?

Mobile devices are prone to failures and to support robust execution of business processes, adapting backend solutions to mobile is necessary. The existing infrastructures companies have developed with stationary computer users in mind simply can not cope with mobile devices.

Regardless of architecture choice, all mobile enterprise applications include mobile client and backend server. There are backend server architectures with business process support or without. The next chapter studies platform and architectural choices for mobile client which conform with challenges stated above and chapters after that investigate methods of integrating mobile into backend servers with business process support.

3 Supporting Mobile Devices in Business Processes

On the backend servers with business process support, there are business process models and business process engine. Business process definitions consist of user interaction tasks among with other business logic. User interaction tasks traditionally designed for stationary computers in mind and therefore, assumes users have reliable internet connection and stationary computers.

Any service oriented environment should allow for mobile device support during business process execution [PMR14]. Supporting mobile devices means enabling users to use their mobile device to participate business process executions which they used to be only allowed to perform on stationary computers.

From the appearance, it is as simple as only adapting computer user interface into the mobile user interface. In reality, there are more to this. Mobile devices have some unique characteristics which make them differ from stationary computers. These add more requirements of optimizing user experiences and handling errors from process definitions. In the case of some business process needs to run on user's mobile, a new light-weighted business process engine is needed on the mobile devices.

3.1 Business process modeling

Any service from providing weather information to most sophisticated knowledge intensive health-care service can be a business process. A business process can be a combination of purely online services or can be real services such as delivering a package or more commonly mix of both. A business process can be as simple as between client and server or can include multiple parties and orchestrated by single coordination node. Note that workflow means business process, they are interchangeable terms in most context [APM03].

Business processes can be defined in modeling language and programmed so they can run on the business process engine. Technologies provide means to design, optimize, manage and analyze operational business processes called business process management (BPM). BPM includes tools, methods, and language to express and execute business processes such as Business Process Execution Language (BPEL) [JEA07] and Business Process Modeling Notation (BPMN) [VHa10].

BPMN 2.0 is UML based modeling language which is in both graphical and XML format at the same time and also easy to understand by business world and technology world. BPMN 2.0 has an executable and non-executable version, executable version is relatively formal and has a higher requirement, but can be directly executable by business process engines as the implementation of the process, no extra coding is required in between.

BPEL has a longer history than BPMN and supported by most of the middleware products. As being a modeling language, BPMN 2.0 has been built on the success of BPEL and solved the visual representation problem of BPEL [Ley10]. Only a subset of BPMN 2.0 is supported by BPEL, though, and rest are still under discussion. On the newly developed process engines, BPMN 2.0 has direct support and does not need to be converted to BPEL in advance. BPMN 2.0 has gained more popularity and becoming an industry standard.

Business process logic is expressed explicitly by using starting and ending point, sequence flow, user task, service task and exclusive gateway elements of the language. Figure 2 shows an example of simple weather information service in Business Process Modeling Notation 2.0 language.

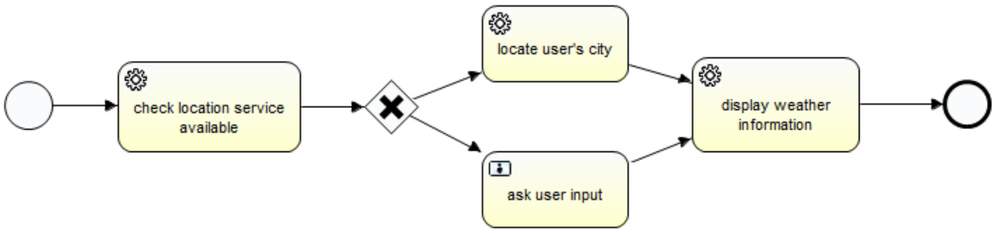


Figure 2: Simple weather information service in BPMN 2.0

Business process can be improved by error handling. Varies errors might occur during process execution and they cause failures. Figure 3 shows a little bit improved version of weather service with some additional error boundary event in case some error happened or cancel end event in the event of user canceled. The obvious advantage of improved version of the business process is it is more robust and less likely to end up with unhandled exception failure. For example, in the first version of weather service, the user interface might just hang forever or end up with a sudden crash of software because of location service failure due to poor GPS signal. In the improved version of the service, the process will switch to the manual input step of user's address in case such a failure, hence improved the user experience. For failing to retrieve the weather information error or failing to get user's input error, the process will also end nicely with a clear error message. Otherwise, when these exceptions occur, process behavior is not defined and users might get confused about what is the underlying error.

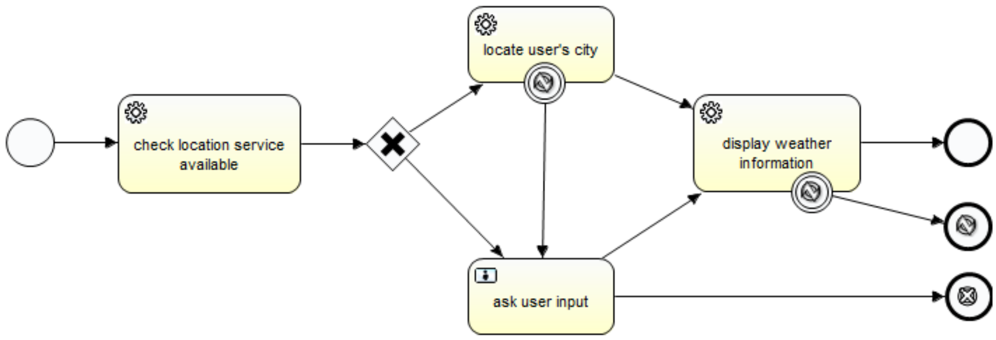


Figure 3: Weather information service with exception handling

3.2 Mobile devices can improve business processes

Mobile devices can improve processes and benefit enterprises significantly. If used correctly, it can simplify business processes and speed up the business process execution, therefore, save cost and at the same time bring market competitiveness. Enterprises who brought mobile into their business processes almost always can run higher efficiency and speed [Lau07] [ZBr99].

Here is one example of how beneficial mobile tasks to traditional business processes. Retail chains have their loyalty card system which helps them to manage their customer relationship. It is a proven method of identifying and growing the loyal customer base by giving them some bonus. There is always particular process they offer when new customers want membership cards. As in the Figure 4, the business process model on the top shows how it was done traditionally before introducing mobile tasks and the one at the bottom shows the new business processes after adding mobile devices.

Traditionally customers have to visit one of the retailer's store to apply for the loyalty card. They first fill in their application on the paper form and pay for the cost of their loyalty card. Sometimes retailers offer them the card for free, in that case, the retailer will take the cost. After the application is submitted, it will be filled into retailer's database by their employees and the order for the new loyalty card will be placed to card manufacturer. The card manufacturer then manufactures the new loyalty card for the customer with their name on and send it to them by

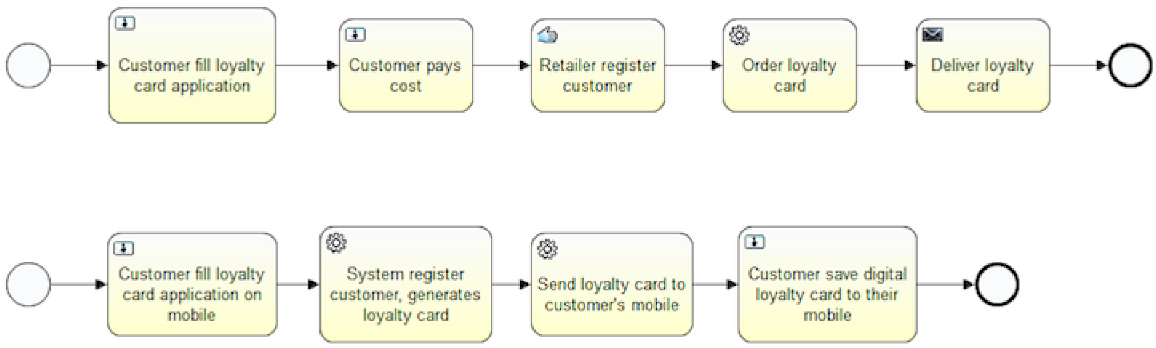


Figure 4: Example of integrating mobile tasks into conventional business process

post. The whole process can take weeks of time until the customer can receive their new card and start to use with additional manufacturing cost and delivering cost.

The new simplified business process allows customers to put their application anywhere from their smartphone by just visiting the retailer's home page. Within seconds after customers input a few details and submit their application, the IT system of the retailer will automatically generate a new virtual loyalty card and send it back to the customer's smartphone. Customers can save their new loyalty card into their smartphone without increasing their wallet size and start to use in immediately. The whole process can be accomplished within minutes without any cost to both customer and retailer.

3.3 Build and runtime environment of business process

Business processes lifecycle include build time and run-time stages. Build time activities include creation, configuration, and verification of business process model definitions. After process definitions are deployed to business process engine, business process instances are created by process engine as per user request invocation of services. Multiple instances of same process definition can exist at the same time. Business process engine will provide the environment for execution of business processes by creating process instances from process models, interpreting and managing them [RWe12]. Figure 5 shows the overall details of business process creation and running process.

Interpreting process models means performing the defined behavior of activities such

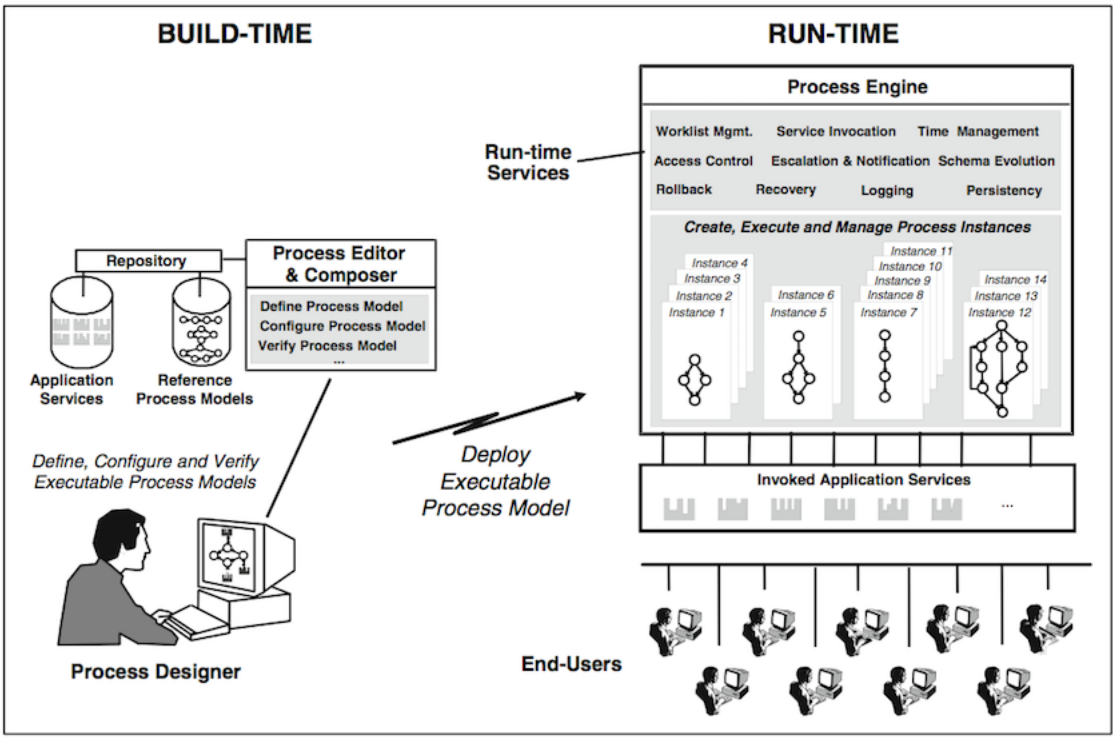


Figure 5: Build-time and run-time environment of business processes [RWe12]

as the invocation of application services using the parameter data. Managing process involves functions such as monitoring the progress of process instances, logging the events.

3.4 Distributed execution of processes with mobile devices

To support mobile devices in business process execution, solutions to distribute business process execution is needed. Distributed execution means supporting multiple server and devices to take part of business process execution, and this is a topic has been researched before mobile devices come into the picture.

Distributing the execution of business processes to multiple devices aims at enhanced scalability. However, it brings the question of when and how to determine which device should execute some part of business process. This is called process coordination. Solutions which decides this at design-time is not flexible at run-time. If problems occur such as random failure of a node, it is hard to recover from the error and continue the execution.

Other requirements of process distribution include process synchronization and process update. Process synchronization means communicating the current status of the running process between participant devices. If the same data element is written by multiple process fragments, synchronization is required to handle data consistency. Process update means process definition updates should be easy to be propagated to all participants of process execution at run-time.

There are three known solutions to distribute business process execution on mobile devices. Namely, process partitioning, process migration, single mobile task handling as shown in Figure 6 [PRB15].

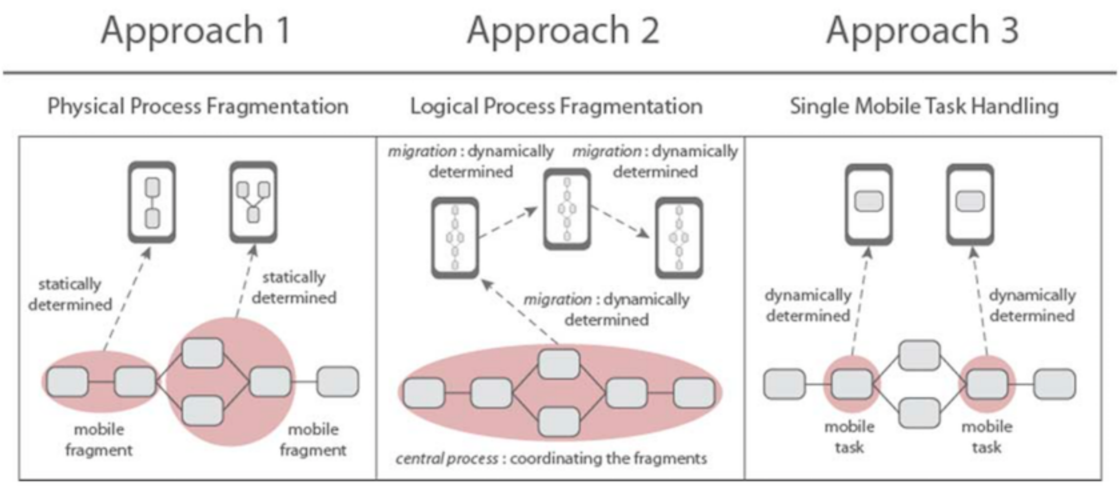


Figure 6: 3 Different approaches for integrating mobile into business processes [PRB15]

Business process fragmentation, which known as process partitioning, is the result of attempting to decentralization of business process execution [ZKK10]. It was not created for mobile tasks. During the collaboration of multiple parties in the business process, participants tried to avoid creating centralized coordination for business process execution for technical and security reasons. Therefore physical partitioning of business process has occurred.

Process partitioning means splitting the business process to several partitions and distributing them to multiple roles in the business process. Since partitioning has occurred before business process instance creation, synchronization among multiple business process engine should be considered at design time. If devices encounter physical challenge during the synchronization, it will be difficult to recover business process execution. When multiple branches of the business process run on different device concurrently, synchronization of them can be challenging [PMR14].

On the contrary, logical process fragmentation, which also named process migration, distribute only the responsibilities of running specific part of business process. By fragmenting business process logically, each participating business process engine still has a full definition of business process and synchronization decisions left to runtime based on the process scheme and availability of mobile devices [ZKK10].

Single mobile task handling assigns a single task to one dynamically determined mobile device and keeps the business process execution in the centralized process engine [PMR14]. Single mobile task handling solution is the most practical and suitable for common use cases because of the other approaches requires business process engine running on mobile devices and that technology is still not matured yet [PTR10].

In process migration process instance migrated between different sites carries all data produced by previous sites and definition of process schema. All the data can be used to determine the subsequent device to continue business process execution and its responsibilities, which make coordination more effective [DRM12]. However, process partitioning method is not preferred due to its drawbacks [DDF08]. Process partitioning method needs to pre-determine execution device before process instance creation, in consequence, difficult to coordinate. If one part of the process fragment needs to be changed, then there is need to propagate these changes using special protocol.

When applying process migration solution to distributed process execution with mobile context, there are mobile specific challenges in addition to the synchronization challenge. These challenges that will be discussed in later part of the thesis are often neglected by current implementations of distributed business process execution.

3.5 Improved process migration

When processes migrated, in order to make the communication more efficient and precise, process description needs formal model to communicate the current state of migrated process instance [ZKM10].

Process Migration Data Meta-Model can enhance original business process definition in Business Process Execution Language for Web Services (WS-BPEL) [JEA07] and the XML Process Definition Language (XPDL) [ZKK10]. Process migration data meta-model is an enhancement fully comply with process definition languages mentioned and does not affect compatibility with existing implementations of process engines. It only helps during process execution provide contextual support for process migration. However, original process definition without migration data will still be able to run as before in centralized mode.

Process migration meta-model assumes process definitions include a finite number of activities that need to be executed. Each activity has a limited number of variables that hold data required by activities. So it includes a migratable process lifecycle state model, which represents execution state of the whole process and an activity lifecycle state model for each activity, which represents the state of the activity. As long as process can be executed in local process engine, there is no need for process migration. Process state *Running* represents process has an atomic activity currently in *executing* state, therefore, should not be disturbed. Process state *Option* is used to represent a stable state which process can be transferred [ZKM10].

In addition to keeping track of the state of the whole process execution, process migration meta-model also keeps the state of data. For privacy and security reasons, storing process state in an external database is not allowed. Process migration meta-model can mark some activities as start activities which will be executed as the first activity after process migration [ZKM10], therefore, each activity needs to have a unique identifier.

Process migration meta-model allows a process to specify its *Selection Type* to specify the strategy of selecting next device which process is going to migrate to. It has several options. The default option is *Undefined* which allows current process engine can make the decision based on availability and resourcefulness of candidate process engines. The option *Fixed Participant or Role* defines specified human, or process engine should perform the process or set of activities. *Algorithm* option allows current process engine to select participant using the specified algorithm. *Variable* option allows process engine to select the next participant from a variable in process description itself. *QoS and Context* option specifies process engine should select the next participant based on some quality of service criteria and some context such as physical location [ZKM10].

All these options enable more dynamic and flexible execution of business processes and help overcome the limitation of communicating the state of running processes. These are also can be used to overcome mobile specific challenges in order to help improve robustness of process execution with mobile devices.

In process migration, process servers need to transfer large amount of data in order to communicate process state. In case there are lots of process servers and they are located far from each other, therefore has low speed internet as mean of connection, this can pose challenges to smooth execution of process instances.

ADEPT project [RBa07] developed technology to allow ad-hoc modification of distributed business process management systems. At the core of their implementation, there is a central control instance, which has the full picture of global state of the business process instances. When the business process schema is changed, it only needs to updated at the central control instance. Central control instance then assess the possibility of updating business process schema in each instances based on their current state. When decided, central control instance will send only latest change history to the device running business process instance because in process migration all the participants have the full definition of business process schema. Their prototype proves the concept of seamless integration between distributed business process management system and centralized ad-hoc business process modification can be achieved with acceptable communication cost.

In case process fragmentation was required because of privacy or security reasons, process migration lacks proper security mechanism to protect from unauthorized access [ZKM10]. Process migration requires all participant to have a full copy of process definition schema to minimize communication and coordination efforts. That results in unnecessary disclosure of process data in some cases such as credit card information and process flows which are internal and valuable for particular enterprises. A method called process masking can be used to protect such vulnerabilities. In short, process masking will use encryption techniques such as PKI (public key infrastructure) and the session key and combine them effectively to encrypt the communication and important parts of the process. That will make sure only desired parties with correct private key have access to the sensitive information [ZKK10].

4 Implementing Mobile Applications

Mobile services include backend and mobile client part. After considering how to support mobile devices in their business processes, enterprises also need to consider how to implement actual mobile user interface.

Before implementing mobile application, one should understand there are multiple ways to develop applications for mobile devices and each of them has some advantages and disadvantages. To develop application with minimal cost and reach widest possible audience, it is good to first review the options.

One of the good old methods of saving software development cost is software reuse. Since there are multiple mobile operating systems exist in the market, and they are fragmented, software reuse between major operating systems can be one effective way of saving mobile application development cost.

4.1 Mobile application development paradigm

Enterprises have multiple architecture choices for developing mobile applications. There are four different methods, namely, native application, web application, mobile widget and HTML5 mobile application [HuV12].

Native applications can be developed using the software development kit (SDK) provided by the operating system manufacturer, and usually specific to that platform and cannot be easily ported to another platform without rewriting the major part of the software. The native application does not mean the application is written in a programming language that compiles into native machine code, such as C/C++. It rather means using the programming language that is natively supported on the platform such as Java in Android case and Objective-C and SWIFT for iPhone.

Native applications are usually compiled and able to leverage the most capacity of the device due to their performance is optimized for the platform they are running on and they are able to utilize all the features platform can offer. It is relatively easy to develop native applications since it is the official way of developing applications for the platform and therefore has good development tools and documentation support from the vendors. Native applications can be distributed via application stores, and users can install updates automatically. Application user interface and user experience are usually consistent with other applications on the platform and relatively familiar for the users. If the application requires rendering of 3D graphics using accelerated graphic processing component, then developing a native application is the only choice [HuV12].

The major issue with native application development is the cost of supporting all different mobile platforms and delivering the application to users through their platform specific application stores. It is almost impossible mission to natively support all the different mobile devices in the market[ChL11]. Usually, enterprises choose typically top 2 or 3 mobile platforms, and smaller companies and startups can afford only one.

Widgets are task specific, light-weighted but also independent applications. They are allowed to run on device desktop without the user performing starting an action. They usually leverage network resource and show real-time updates such as weather or currency exchange information. Widgets have very limited functionality, therefore, can not be equal choice comparing to other application development architectures.

Mobile web applications require nothing to be installed on user's devices. Users use mobile web application via web browsers on their mobile devices. Mobile web applications are the mobile adapted version of the desktop web service. Usually they are adapted to smaller screen size and mobile user experience and requires less amount of data transfer [MHL10]. However, because of current technological advancement in HTML5 and mobile browsers, users almost can get seamless user experience as mobile native applications [HuV12]. Mobile web applications can reach broadest user base because every mobile device has a browser as the standard feature. Mobile web applications can be updated almost without user notice on the server and enterprices can save complicated process of delivering the application to each platform. For services purely deliver content, the mobile web application is the best choice due to their cost-saving and broadly supported characteristics.

Web applications perform differently on user devices depend on the degree of HTML5 support on their browser. Even though HTML5 is published by W3C as a cross-platform solution, in reality, developers face problems of cross-browser compatibility because all the platform interpret some HTML5 APIs differently [HuV12]. Therefore, it is very complicated to develop a functional and robust mobile web application.

The HTML5 mobile applications, also known as hybrid applications, are the applications developed using HTML5, JavaScript, and CSS technologies and packaged with the help of cross-platform development tools usually provided by a third party. This technology is possible because almost all the mobile devices have a browser application and it is programmatically possible to instantiate a browser instance and interact with its JavaScript interface from native code [ChL11]. However, because of Hybrid applications render application UI and logic through the help of browser engine and JavaScript instead of directly writing pixels on device's display as native applications do, users should anticipate somewhat slower UI performance. However, the advancement in JavaScript engine performance made that difference is rather negligible for most of the light-weighted applications [ChL11].

The hybrid applications can share code between different mobile platforms. They need to be compiled and packaged using third party tools to be delivered to application stores such as PhoneGap. They will have different look and user experience than the applications native on that platform but will have a consistent user interface across different mobile platforms. The hybrid applications have the same browser fragmentation challenges mobile web applications have, thus, ought to be tested on each platform for compatibility. Distribution and updates go through application stores as native applications do.

One recent evaluation [KKR15] of those cross-platform development tools shows that PhoneGap is the most suitable platform for cross-platform development. The other competing frameworks either lack some necessary API support to sensors or one important operating system. PhoneGap has outstanding performance, documentation and community support advantages. It is open source and free to use for commercial purposes as well.

In one experiment [KKR15], the natively developed Android application with the same design and purpose compared to the cross-platform version of the application developed using the PhoneGap. The cross-platform version has only slightly increased application startup time within a very acceptable range, and package size increased about 50% which is also fine considering the current devices have a very vast amount of storage capacity. However, RAM requirement is more than three times as much and even though current mobile devices have gigabytes of memory, but for more complex applications that also target middle and low range smartphones, this could pose some problems.

It is worth to notice there are also commercial third party frameworks exist nowadays which are different than traditional open source frameworks. They allow users to code in a language other than HTML and JavaScript to develop cross-platform applications and even has native graphical support. According to their official websites, the Xamarin [Xam15] lets users develop cross-platform applications in C# that can be compiled into native code on each platform. It also offers solutions for testing application on their test cloud virtually on 1000 devices for compatibility. Qt [QtA15] even provides solutions for developing cross-platform 3D graphical applications in C++. There are few academic papers available from credible sources about those platforms, and those platforms come with a good price tag. So they are not evaluated in detail here.

Hybrid applications aim to reuse application code developed in HTML5, CSS and JavaScript. They are wrapped around a native container that facilitates access to native device features. Because of cost efficiency, this architecture is getting popular among mobile business applications. According to Gartner analysis by 2016, more than 50% of mobile Apps deployed will be using hybrid architecture [Gar13].

	Native	Hybrid	Web
Processing Power	High	Medium	Low
Access to Local Resources	High	Medium	Low
Connectivity Requirement	Low	Low	High
Code Sharing	Low	Medium	High
Development Complexity	Low	High	Medium

Table 1: Comparison of Mobile Application Architectures

Table 1 compares different mobile application architectures. The left column lists each criterion and the other columns on the right show the characteristics of three optional architectures in relation to the criteria. Native applications can leverage most processing power as they are compiled into the most optimal binaries for CPU than other architectures and can directly program GPU (Graphical Processing Unit) which other architectures cannot. Hybrid applications are at the medium level as they can still run in the local process within the native container and web applications need to offload processing to servers. Native applications have full access to local resources such as sensors. Hybrid applications can access only part of the local resources depending on the middle layer they are built on. Web applications have the least access to local resources [SSP13]. Native applications and hybrid applications can work offline. So, they have the lowest requirement for connectivity, and web applications always have to be connected. Native applications can share code only within the same mobile platform. Most part of hybrid application codes meant to be shared among multiple platforms and web applications should work regardless of the mobile platform they are running on. Only HTML5 support in the browser application on the platform is important for web applications. Development complexity should be the simplest in native applications because they are the official method of developing applications and usually supported by quality tools and examples. Hybrid applications are the most complex since they have an additional cross-platform layer. Web applications also deal with the similar cross-browser complexity.

4.2 Reusable mobile business process applications

Understanding the requirements for an application is often the most costly part of the project. Since mobile application developers have little knowledge of the domain for which they are implementing the application, significant amount of time will be spent on communication between developers and domain experts to understand the requirements. In standard development practices, there will be a lot of iterations during product development for even smallest changes in the requirement. It could be costly to hire a full-time mobile application developer for small business. Sometimes it is easier to teach mobile application development knowledge to a domain expert rather than teaching a mobile developer domain knowledge thoroughly.

After application is developed, it enters long maintenance lifecycle. During that time, there will not be big changes to the application. However, rather small and continuous adaptation of application logic is needed from time to time due to changes in the process. It is rather costly to hire a application developer for maintenance for such a long period of time.

In order to save high maintenance cost of mobile application after its development, it would be great if domain experts can reconfigure it without mobile application developer's help if needed. Even better if application logic can be expressed in a more visualized modeling language which domain experts can understand and maintain by themselves. Mobile application can be developed in a way that consumes such model and directly customize application's user interface.

Process-driven mobile applications [SSP14] try to generalize mobile applications and isolate the domain specific knowledge from the mobile applications as much as possible. Therefore, ideally, one application can be used for multiple business purposes, and only business process definitions need to be maintained by experts.

One use case of such architecture is data collection applications [SRP13]. Very often educators, health care industry or marketing departments want users to fill some questionnaire to collect data or getting feedback. Filling paper-based survey and inputting them into a database is inefficient and yet developing a mobile application for each of this new questionnaire is a burden. This type of applications are simple and have similar architecture, they just require users to answer some questions and sends data to the database for storing and analysis. Probably after the user finishes answering all the question, they give users one initial feedback. Only variation point is differences in questions and their form. Because of similarity in architecture, this type of applications easier to be generalized and business process can be separated from application implementation. Johannes et al. [SSP14] just achieved such proof of concept.

In the new architecture, questionnaires can be modeled using business process models described in Business Process Modeling Notation (BPMN 2.0). Questionnaire decision points can be expressed using gateways and each question can be one activity and can be displayed using one page on the mobile device's screen. Pages can be switched according to users' selection and definition of the business process.

Business process engine will run on a remote server. Users need to select which set of the questionnaire they are answering at the beginning, so the corresponding business process will be picked to run. Mobile device communication with process server upon each user interaction and page swap. When the mobile device receives the definition of the process model, it generates UI for that question automatically and displays.

With this architecture, one mobile application can be used to facilitate as many questionnaires as needed. The mobile application needs to be deployed only once, and questionnaires can be described in BPMN 2.0 and maintained later on. Domain experts can be trained to use BPMN 2.0, and it is much easier than developing a mobile application for each questionnaire. Experience also indicate that users are more willing to answer surveys by an intuitive mobile user interface.

However, since process-driven mobile application architecture [SSP14] is still in prototype state, during the experience, several shortfalls are also discovered. It seems current BPMN editors are not intuitive enough and there is a need for a better user-friendly domain specific modeling notation and tools. Also because application user interface is automatically generated, the possibilities for customizing the layout is rather limited. Since all process models are stored and executed on the server, at least a stable internet connection is needed on the mobile device. To make the application work also in the offline state a light-weight process engine is required on the mobile device.

4.3 Middleware based approach

The MUIT (Mobile User Interactions and Tasks in WS-BPEL) architecture [LXH16] has taken the software reuse to the next level. By introducing middleware between process engine and mobile devices, the MUIT middleware architecture can automate generation of user interface for mobile tasks.

Software reuse can be implemented at multiple levels. As business applications have similarity and variation in some layers, software reuse for different purposes with small modification is possible. Automated generation of user interface even takes software reuse to the next level.

For the business processes that contain mobile tasks, there is a middleware-based approach [LXH16] to generate the user interface for mobile task execution. Traditionally business processes supported by native applications on mobile which will be time-consuming to develop and maintain. The MUIT (Mobile User Interactions and Tasks in WS-BPEL) architecture solved this problem

The mobile user interface for each task is described in MUIT DSL (Domain Specific Language [KSV09]). On the run time, MUIT service engine will generate an adaptive UI in HTML, CSS and JavaScript code based on the mobile platform and screen size of the user device and send to the user's mobile. The web browser on user's mobile then renders the UI to the mobile device's screen.

At design time, after process model is created in BPEL, if an activity in process model requires a user interface it is modeled as MUIT service with a reference number. MUIT developers also need to specify the UI in MUIT DSL as in Figure 7, then process model is deployed on business process engine, and MUIT runs on MUIT engine as standard web service. At the runtime, when the process needs user action to complete, an MUIT service request send as SOAP request. MUIT server then sends a notification in user's preferred format either as SMS, instant message or e-mail and wait for the user to confirm. When the user confirms the request by clicking the hyperlink, browser application creates HTTP request to MUIT server and user interface is presented. During task execution, mobile and MUIT server keeps the communication in JSON format.

4.4 From existing web-based enterprise applications to mobile applications

Reusing the existing IT infrastructure components is another way of saving cost. Since many enterprises have web based enterprise applications, developing mobile client does not have to start from scratch. Even though current web interfaces most likely can not be reused directly in mobile web interface due to their complexity, but already developed business logic and many other components can be reused while

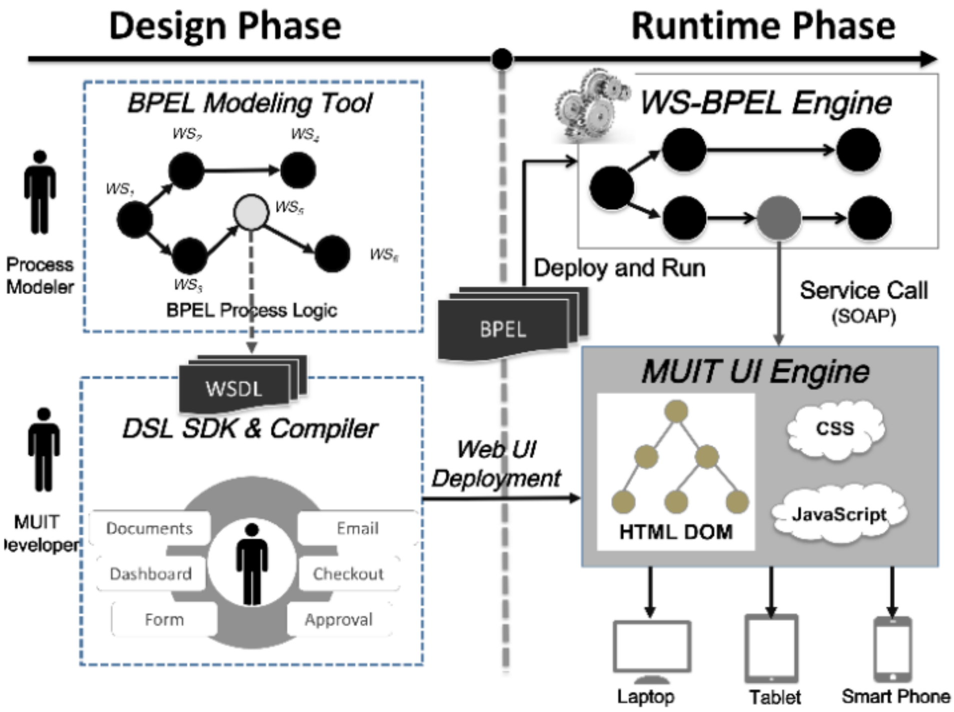


Figure 7: The MUIT middleware [LXH16]

developing mobile application support.

Most enterprises have their existing web-based applications that serve their customers. Even though it might be old, but it already includes a significant amount of company's internal processes, therefore, is a valuable asset. It is also a good starting point for enterprises to develop mobile application interface. By reusing it correctly, enterprises may save development cost.

There is a strategical method for developing mobile applications from web-based enterprise systems called Metamorphosis [IGi15]. Activities during development can be divided into 4 phase, requirements, design, development, and deployment, as presented in Figure 8.

During requirement phase, feature requirement of the mobile application is collected and validated with the stakeholders. The existing set of features from the web based application are identified and mapped to possible candidate features in the future mobile application. There is also need to evaluate the feasibility of planned features on the mobile platform. Since not all the functionalities are possible to implement on mobile due to limitation, some selection and adapting is needed [IGi15]. Some examples of adaptation could be simplifying the forms need to be filled by users since

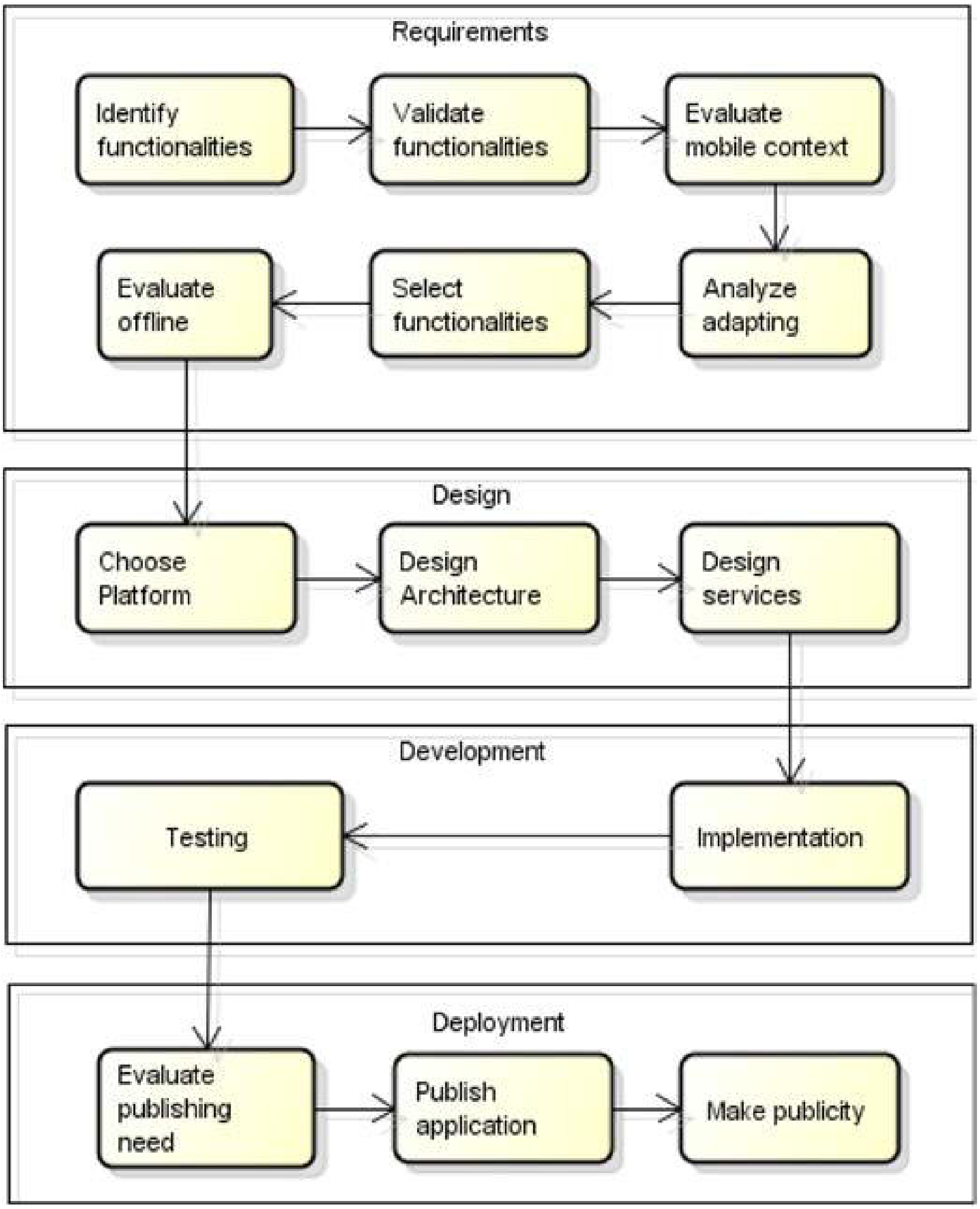


Figure 8: Metamorphosis [IGi15]

mobile users have lot smaller screen and require intuitive user experience [FJA12]. Making operations also work offline is also useful due to mobile networks are not always available.

At the design phase, selection of mobile platform should have been made. The section of mobile application development paradigm has introduced different available choices and their pros and cons. It is a decision between the web, native and cross-platform applications. If the native application has been chosen then next choice would be the mobile platforms to support.

After choosing the platforms, then designing system architecture and services is the next step. Integration of new mobile application with the existing system, utilizing the existing business component as best possible way should be considered. Web systems are usually layered architecture, and its user interfaces logic (front-end) is separated from its business logic (back-end). The mobile application could be integrated with its business layer and utilizing its stable core. The mobile application retrieves data from existing web services through its REST APIs from business layer [ZKo13], and offline operations should also be taken into account. Different degree of refactoring existing business layer code still needed because first, it has not been considering mobile application interface when it is designed; second, over years of maintenance, there are usually some violations of software architecture principles perhaps happened, and the separation might not be so clear as it should be.

The architecture design of the mobile application should be completed at this stage which includes tools for selection, design patterns and programming language, etc. Once above steps are ready, the requirement of running offline can be verified and after that development can be started.

Implementation and testing should belong to the development phase. The actual source code of mobile application and Internet service will be written and tested all functionality works as expected. This stage could be many iterations of the minimum viable product to complete product. As it always improved based on scrum and agile processes to make sure implementation meets the product requirement.

Deployment phase will be the different according to the choice of platform. Web applications can be directly published, native and hybrid applications need to go through platforms publishing and review processes. There is also the possibility to deliver the application through organization's own application store. By this way, it will reach its target audience via trusted channel, and it will stay in a relatively private. Only publishing the application is never enough, there should also be enough communication, advertising through a launch event is needed to target audience get informed.

4.5 Summary

Table 2 summarizes the recommendations to reduce development costs for mobile applications. Savings can be achieved by reusing the assets in multiple levels. In code reuse level, sharing code among multiple platforms should be the target. The

Code reuse	Selecting the right application architecture that suite requirement
Component reuse	Repurpose existing web applications into mobile
Application reuse	One application for multiple purpose, separating process from application and generate user interface dynamically within mobile application
Code generation	generate web mobile interface within middleware

Table 2: Approaches to Reduce Development Costs for Mobile Applications

operating system world of mobile is fragmented by different hardware and software vendors in the market. Supporting every type of mobile device natively is an impossible task to achieve. Companies need to be selective in their target audience and platform. There are native, hybrid or web architectures to choose according to the requirement for application performance, target audiences and cost limitation of the project. By using the web or hybrid mobile application development approach whenever possible, companies can achieve saving cost by supporting multiple platform using the same code.

In component reuse level, companies can convert existing web-based enterprise services to the mobile application service and save by reusing the old components for new purposes. In the application reuse level, there are techniques such as developing one mobile application for multiple purposes by separating business process logic from the application interface.

In code generation level, using middleware-based approach to generate mobile web interface directly from process definitions can totally skip developing the mobile application.

5 Adaptive and Flexible Management of Mobile Process

Mobile services are gaining increasing popularity in every domain and used for mission critical tasks such as health care and aviation. At the same time, they are prone to failures because of user habits and mobile specific limitations such as limited battery life, unstable internet connectivity. To prevent single task failure does not cause catastrophic consequences or failure of the whole process, certain exception handling mechanism should be built into business processes. Business processes should not assume users will use stationary computers with reliable Internet connection and they should take mobile users into account as well.

Adaptive and flexible process management technology provides a solution for flexible execution of business processes and therefore support for proper handling of runtime exceptions. In this chapter, we introduce adaptive and flexible process management technologies.

5.1 Mobile specific challenges to business process

Section 2.4 described a mobile shopping experience and showed how challenging it can be. This section investigates the mobile challenges of business processes behind that story from implementation and technical point of view. Challenges are main differences from conventional business processes and their implementations that must be considered while integrating mobile tasks. The mobile challenges can be in the environment (e.g., unstable connection due to weak signal), or in the process (e.g., missing data because of process failure) or user related (e.g., instant shutdowns of the mobile device) [PRB15].

Connectivity is the first challenge related to the environment, it refers to the situation of candidate mobile device is not connected to the network for any reason. For example, the device is broken, or user is off from work. That also include unstable connections because of the user is at low signal area and connection can be sometimes established and not at other times. Only mobile devices with stable network should be considered as a candidate to run business process [PMR14]. For mission critical tasks this also should be taken into consideration.

In case the mobile device has low battery and is therefore prone to disconnect at any time. This could happen to all mobile devices regardless of their battery life.

Furthermore, Users might shut down their devices during business process execution which surely disrupts the business process execution [PRB15]. This can also happen if users switch to different applications because of there are instant messages or push notification from other applications caught their attention. As a result, the application which is running business process goes to background. Most mobile operating systems treat background processes as going to suspended because they would like to devote most of the precious system resources to new foreground application and also to save battery [IBE16]. This situation might also happen involuntarily if the application takes relatively long time to complete some tasks and does not require user action during that. The device goes into sleep mode automatically due to user's inactivity for the same reason of saving power. It is possible to continue run mobile tasks in the background in such situations for a legitimate reason, but that requires special care during application development and deployment process.

Some mobile tasks require users to be at a particular location [PMR14]. For example, they need to be at a particular point to collect data which otherwise wouldn't be available to them. The location requirement is both process and user related challenge.

User's attention and urgency of the mobile task [PMR14]. Some mobile tasks might be very urgent and must be finished within the defined amount of time. This process related requirement constitutes a challenge if a user is not paying attention to their mobile device or their mobile device is silenced off. So for this very urgent mobile tasks, some other protocol should be in place.

After integrating mobile into business processes, some process steps will become data dependent to the mobile task [PMR14] and can not be continued until data is realized by the mobile task.

Although these challenges are valid at some point, they don't necessarily apply to every project at the same time [PMR14]. Every individual business process must be analyzed separately to identify which challenges are relevant. For example, urgent mobile tasks are probably necessary in business processes in hospitals and not needed as much for other cases. Same as location requirement is not so relevant to the questionnaire or survey type of business processes.

5.2 Backup and delegation service

Due to the mobile-specific challenges described in the previous section, it is important to make a receipt for mobile related failures during distributed execution of business processes. Enterprises should introduce more error recovery methods to their business process execution architecture when they integrate mobile devices. There are multiple solutions of error recovery methods from failures in business process execution failures.

Physical process partitioning has been proposed as a method to robust execution of business processes on mobile devices [BMM05]. It will alter the execution model to generate isolated partitions in order to minimize the synchronization effort between execution devices.

Backup and delegation service technique [PMR14] introduced error recovery methods to business processes to achieve self-healing from exceptions. In short, the delegation service determines the subsequent mobile device to perform the specific task in case of an error. When a mobile user has not been found after iterating all possible candidates, the backup service will ensure the failure will not halt the overall execution of the business process.

Figure 9 depicts mobile task runtime architecture. At the center of the new architecture, there is Mobile Execution Environment (MEE) as an extension to existing business process engines which does not have mobile task interface yet. Mobile execution environment maintains a list of mobile devices to use for mobile task delegation and backup service. Besides, it has two primary interfaces. One is engine independent process management interface to integrate with business process engine and the second one is the mobile interface to integrate mobile devices to MEE [PMR16]. MEE uses process management interface to get the list of mobile tasks from the process engine, and after performing the mobile task, it will return the data from the mobile device to process engine. All the error handling, delegation and backup operations are handled internally within MEE. The mobile interface also includes another component on the mobile device called mobile service client. This is used for performing mobile tasks on the mobile as well as getting the mobile device status information such as battery status from the device to select the right device to perform the mobile task.

Task delegation works in two patterns. One is transferring rights user to user [CKh08], and another is from the system to user error handling. In the first pattern, part or all of the user's authorization right can be transferred to another user using delegation either in the scope of currently running process or permanently. This function is useful when a user, who has the required rights but restricted by other factors, wants to transfer rights to another user who has the resource. The second pattern allows re-delegation of the task by the system to another user when the original user can not handle the task due to an error [PMR14].

Three separate user lists are used to effective management and reliable execution of mobile tasks. The initial list received from process engine ul_{init} is the list of users authorized to run particular mobile task t_{mob} within the u_{mob}, a u_{mob} list of all mobile users registered in MEE. Using the algorithm in Figure 10, MEE will calculate the list of mobile users ul_{mob} the task t_{mob} can be delegated to [PMR14].

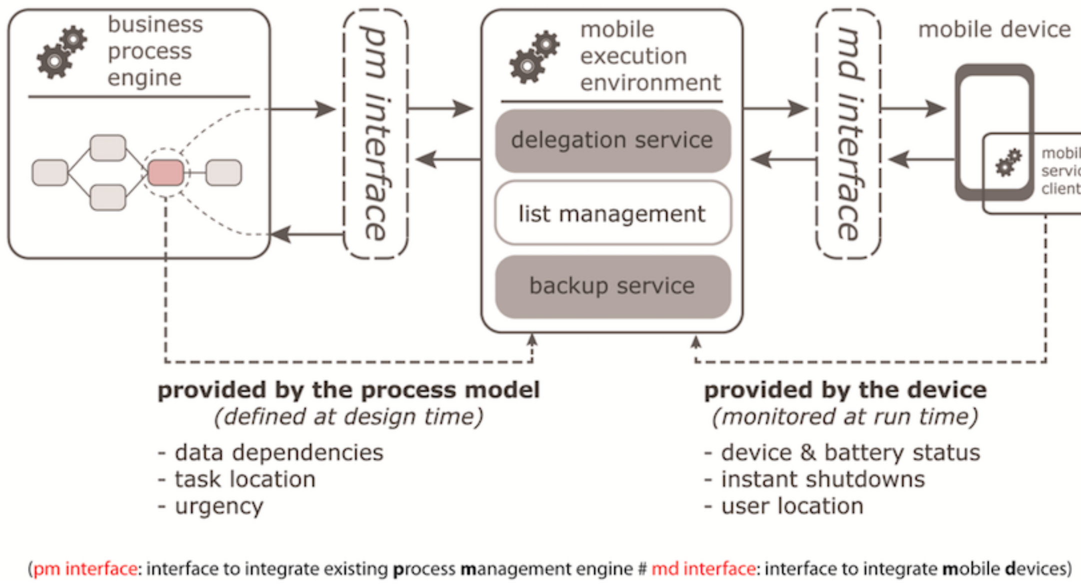


Figure 9: Mobile task runtime architecture [PMR16]

$ul_{mob} \leftarrow \{\}$

FOREACH u_{mob} **IN** ul_{init}

    IF (u_{mob}*.connectivity)* **AND** ($\neg$ u_{mob}*.low.battery.status)* **AND** ($\neg$ u_{mob}*.pre-filter)*

        IF (u_{mob}*.location* = t_{mob}*.location) or* (u_{mob}*.location* = *{} and* t_{mob}*.location=0)*

        THEN

            ul_{mob}*.append(*u_{mob}*)*

Figure 10: Delegate list calculation algorithm [PMR14]

According to the algorithm, delegation list ul_{mob} is the list of mobile devices among which from the common part of the two lists u_{mob} and ul_{init} and also meets important requirements such as connectivity, battery status, authorization, and location. Note that location requirement is optional, if there is no location requirement, then it will not be checked. It is evident that the delegation list will be used to ensure the mobile task will be delegated to devices which only eligible and capable of running the task. After the task is assigned to the first mobile user in the list, all the other users in the list will be used as fallback delegation list [PMR14].

The backup service is a stationary backup task will be run as a last resort when the mobile task is not able to run even after the delegation [PMR14]. Apparently, not every mobile task can be safely replaced by a stationary task. But assuming if the task failure happened due to mobile-specific challenges such as connectivity, then the stationary task which has more reliable connection and operator should be able to handle it. Normally, the backup task will be skipped if the mobile task succeeds through delegation. The delegation will be skipped if the mobile task succeeds at the first attempt as well. Therefore, these delegation and backup services will improve mobile task execution reliability without impeding its speed and performance.

There are simple and complex backup operations [PMR16]. Simple backup operation iterates all the mobile users in delegation list and then calculate the user list for the backup task, and iterates that. In that case, sometimes the same mobile user will be eligible for the backup task which already iterated in the first round and proven to be not working. The simple backup operation still has to assign the backup task to those devices until it is fallback to the stationary user which will be a waste of time. Complex backup operation optimizes this and uses concurrent execution to speed up the execution. In complex backup operation user list for the backup task already calculated when the mobile task starts execution. Mobile task and the backup task will be assigned to the same mobile user at the same time, and if the mobile task fails, user list for the backup task will be updated at the same time. This way, when the delegation list for mobile task iterated, user list for the backup task will also be updated and there will be no more redundancy. This enhancement will speed up the user assignment of the backup operation.

5.3 Robust execution of business processes with mobile

With the help of delegation and backup services, it is possible to greatly enhance the robustness of business process execution regardless of mobile challenges exist. Those challenges need to be addressed during four phases of business process lifecycle: design, instantiation, activation and delegation time.

Figure 11 shows mobile-specific challenges and their possible addressing time during process lifecycle. For example, connectivity problem can be solved by checking mobile user's connectivity state at mobile task activation or delegation time just before the task is assigned.

During the design phase, the original business process will be transformed into a mobile one. Those tasks which are more suitable for mobile users will be identified and changed to the mobile tasks. Location, urgency and threshold requirements will also be added to new mobile tasks if they are applicable. Task dependency will be checked to find out the backup task for the mobile task. The backup task and

Challenge	Design Time	Instantiation Time	Activation Time	Delegation Time
Connectivity	✗	✗	✓	✓
Low Battery	✗	✗	✓	✓
Instant Shut-Off	✗	✗	✓	✓
Location	✓	✗	✓	✓
UserLocation	✗	✗	✓	✓
Data Dependencies	✓	✗	✗	✗
Urgency	✓	✗	✗	✓
| (✓) : is evaluated | (✗) : is not evaluated |				

Figure 11: Mobile challenges during process lifecycle [PMR14]

validation task will be added if needed [PMR16].

While instantiating process instance, user list for mobile tasks is calculated. Only online and free users are considered. Location and urgency information rechecked [PMR16].

During each activation and delegation time, mobile users will be filtered against connectivity, battery status, user habit of instant shutdown and location requirements to avoid failures related to those challenges. If a user has a history of instantly shutting down mobile phone frequently in the past, he or she may not be assigned any mobile task [PMR16].

During delegation, there is also a scenario that a failure in users mobile might cause the whole process to halt for a very long time. An estimation of task execution time will be added to each mobile task [LWR14] to detect the errors promptly and prevent a deadlock situation. While doing the delegation, that value used as timeout and if the mobile user does not return service request in time, it is considered as a failure.

Delegation and backup service method relies on the assumption that multiple mobile users are capable of performing the same mobile task. It did not consider the situation that only single person is capable of doing the mobile task or the mobile task itself is very personalized. For example, a task of giving medical advice to the patient through his or her mobile device most likely can not be delegated to another person. Another example is a payment step at the end of the internet shopping process must be completed by the user who initiated the order. In these cases, different kind of business process execution technology for mobile devices is needed to improve robustness.

6 Supporting Knowledge Intensive Processes on Mobile Devices

The delegation and backup service used by mobile task runtime architecture [PMR14] can improve robust execution of mobile tasks during business process execution. It is suitable for single mobile task handling approach of realizing mobile business processes. It is not suitable for business processes that requires to run on mobile devices for extended period of time. For them process migration method should be used.

The special treatment of mobile tasks from a logical partition of the business processes requires more effort than a single mobile task. Transforming the existing business processes into mobile business process leaves more complexity to business process engine and its mobile execution environment extension. Sometimes process execution requires to stay on user's mobile for longer than single task and expects to run a fragment of business process on the mobile. In this paradigm, mobile devices are more "equal" in distributed execution and assigned a section of the business process. Mobile devices are integrated seamlessly into the business process execution. In consequence, mobile devices need to be equipped with at least a lightweight business process engine for them to handle the process execution as natural.

6.1 Business process complexity spectrum

Enterprises have small to big scales as well as their business processes. The implementation architecture of business processes are very different depends on the process complexity. For example, an appointment system of single barber shop can be implemented using simple client-server architecture and without a need of business process models. If it a service consists of several players and business process definition, but the logic of the business process and roles of each player can be clearly defined, and they do not change easily over time, this process can belong to simple and repetitive business process category.

There are two types of business processes based on their characteristic of execution and maintenance requirements as in the Figure 12. One is called pre-specified repetitive business process, and other is knowledge intensive business process [RWe12]. The simpler transactions which do not really need process definition and process engine called singular services.

Pre-specified repetitive business processes are simpler processes which logic and

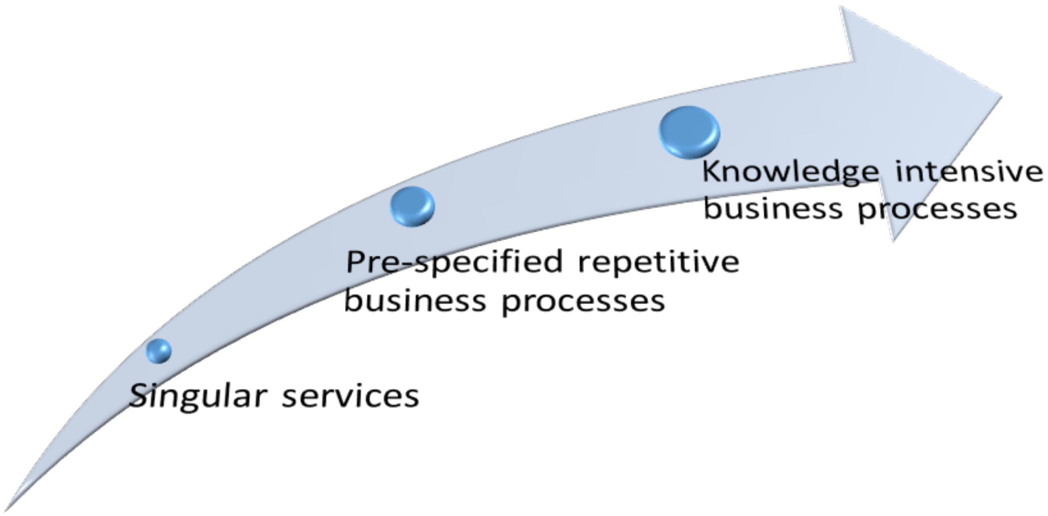


Figure 12: Business process complexity spectrum

behavior are predictable at process design time. These processes simply repeat over and over, and external variables do not influence process logic. All the business process examples mentioned in the thesis so far belong to this category. In defiance of their simplicity, demand for this type of processes is growing and they are very useful in every domain.

On the other side of simplicity spectrum of the business process, there are knowledge intensive processes. They are very dynamic, so it is not possible to select the activities and logic in advance at design time. The process participants will decide them at runtime. For example, there is no single process can be called "Patient Treatment Process" at hospitals that run automatically. Every patient is unique. Physicians and doctors will decide what kind of treatment plan patients have to receive and even after the initial diagnosis and examinations treatment can still be changed following patient's condition. Sometimes there are urgent treatment activities because of dangerous situations. It is generally agreed that such business processes cannot be fully automated [RMa07].

All the business processes captured as process model using modeling language at the first place to be deployed on the business process execution engine. Redesigning and deploying the process model whenever there is a small update on process logic is an extra burden that slows down the speed and it is not always even possible.

Transforming new ideas into services in the fastest speed, adapting ever increasing dynamic market situations is the greatest challenge in today's enterprise's face [DVW05]. It requires greater ability from information systems to cope with the changes. Process-aware information systems (PAIS) have emerged to provide more dynamic and flexible support to business processes.

Knowledge-intensive processes can be partially automated using loosely specified process or data driven process. Loosely specified process does not define the full logic of the process but defines only the known part or the beginning of the process. Activities or series of activities can be automated defined as process model fragments and stored in activity database and made available to actors during process execution. The final structure of knowledge-intensive process mainly decided by the actors who participated in the process [RWe12].

Figure 13 depicts a constrain-based process model example of a loosely specified process which adopted from [RWe12]. It is fracture treatment process in hospitals. It starts with initial activity *Examine Patient* and continue with optional activity *Perform X-rays*. Physician can omit X-ray examination if no fracture is found during initial examination. In that case only *Prescribe Sling* activity may execute because *Perform Surgery*, *Prescribe Fixation* and *Apply Cast* activities must precede with *Perform X-rays* activity. If surgery is performed, the physician is advised to execute *Prescribe Rehabilitation* activity as well. The doctor may run *Examine Patient* and *Prescribe Medication* activities at any time during the treatment.

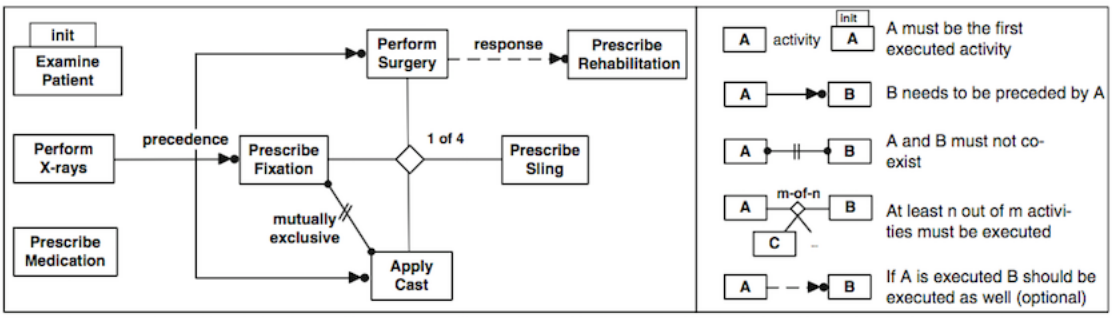


Figure 13: Example of constrain based process [RWe12]

Constrain-based model only captures the known rules among activities about what is possible and what is undesired. As pre-specified process model, constrain-based model is also activity-centric. In some knowledge-intensive processes, even the activities are difficult to identify. In such processes, data produced process is the main driver of process execution. This knowledge-intensive processes can also be modeled using data-driven process model.

Modern IT systems that can facilitate pre-specified or knowledge intensive business process build and runtime environment called process-aware information systems. The build-time environment includes tools to define, configure and verify the process model. Runtime environment means business process engine which has the functionality of creating, executing and managing the instance of business process models [RWe12].

6.2 Mobile business process engine

For realizing knowledge-intensive processes on mobile, single mobile task handling approach is not enough. It is better to use logical process partitioning in that case because not only the single task but more complex part of the process is going to run on mobile devices.

Continuing the treatment process example from the previous section, if a patient has been sent to the home for home-care, traditionally the only way to monitor patient's status is to schedule regular visits to the clinic. It is highly improved if some part of the process run on patient's mobile device can send health data to the clinic while the patient is at home. The physician can give feedback and change the treatment plan if needed. It requires parallel execution of business process both on doctor's workstation and on patient's mobile. This mechanism can be seamlessly integrated with doctors working environment, and the doctor should be able to remotely adapt treatment plan. This kind of use case belongs to knowledge-intensive process because it is highly flexible and can not be pre-specified.

The MARPLE (MAnaging Robust mobile Processes in a compLEx world) Architecture [PTK11] depicted in Figure 14 provides such PAIS that has the possibility to integrate with mobile. In addition to this, it also provides the mechanism to overcome mobile-specific challenges to some degree such as communication failures. The MARPLE architecture consists of two parts, mediation center which extends the process engine on the host and mobile engine runs on mobile devices.

Inside mediation center, maintenance component installs mobile engine to the mobile devices and configures them. Control unit assigns process model fragments to mobile users, activates them, propagate ad-hoc changes and reassign process instances from mobile to mobile. Process instances are initiated either by the mobile user themselves or by the control unit. Process reassigning capability is useful in cases like users want

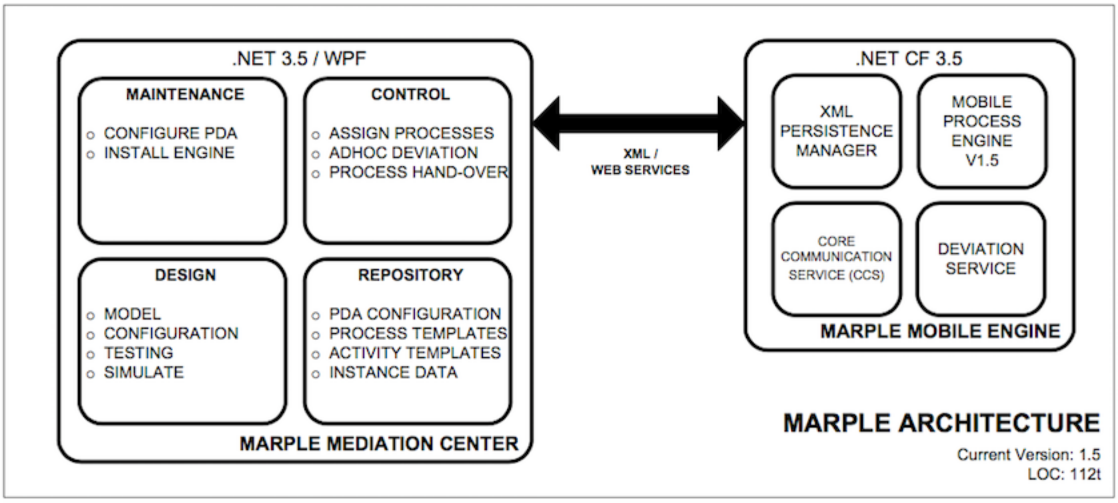


Figure 14: The MARPLE architecture [PTK11]

to switch their mobile device due to battery power or turn to some other person for help due to they are occupied as well as for error recovery on control unit because process execution failed on first mobile device [PTK11].

Modeler component provides tool for partitioning global process into fragments, allocating fragments to mobile devices, linking the activities into the process. This can be used to define the process fragment and assign to patient's mobile as in the previous example [PTK11].

MARPLE mobile process engine has Core Communication Service (CCS) to keep communication with the central unit in the background. XML persistence manager serializes and stores process models for communication and execution.

MARPLE mobile process engine supports parallel execution of the business process on the mobile device itself. It is a useful feature as in the previous scenario, physicians can define parallel activities such as monitoring patient's condition while sending data to the clinic to monitor patient's condition at real time.

MARPLE mobile process engine's deviation service still do not support remote ad-hoc changes to process fragments yet. However, it can propagate local ad-hoc changes to process fragments by adding or removing single activity operation. Integration with calendar application allows scheduling events more visible and easy to update.

6.3 The MEDo project

This section reviews a example project of supporting medical ward rounds through mobile task and process management by the similar group of people who developed MARPLE architecture [PML14] called MEDo (MedicalDo) project. The aim is to provide a real world example, and proof of concept of realizing knowledge-intensive processes on mobile devices as well as to learn from their experiences and validate the theories in real world practice.

Ward round at hospitals is important part of reviewing patient's situation and planning patient's care. It is an opportunity to meet patients and listen to them and patients can learn healthcare staff. The project determined typical ward round is conducted in four steps. In preparing step ward round team gathers patient data and review it. Data is mostly paper based forms and carried in a ward round trolley. Before the patient visit, ward round team discuss patients situation. During the patient visit, the team perform physical examination if needed, discuss patient's situation with them. In the post-processing step, the team reflects ward round and plans for further treatment [PLR13].

The study project has been conducted in seven phases depicted in Figure 15. After analysis of ward round practices, the study found that frequent changes to the treatment plan can occur during the ward round. This requires flexible process support. Current established practice at hospitals is using pen and paper to create task worksheet. Although it is efficient method, it is not formal and loose of information can happen many times due to communication or memory loss, copying or even bad handwriting.

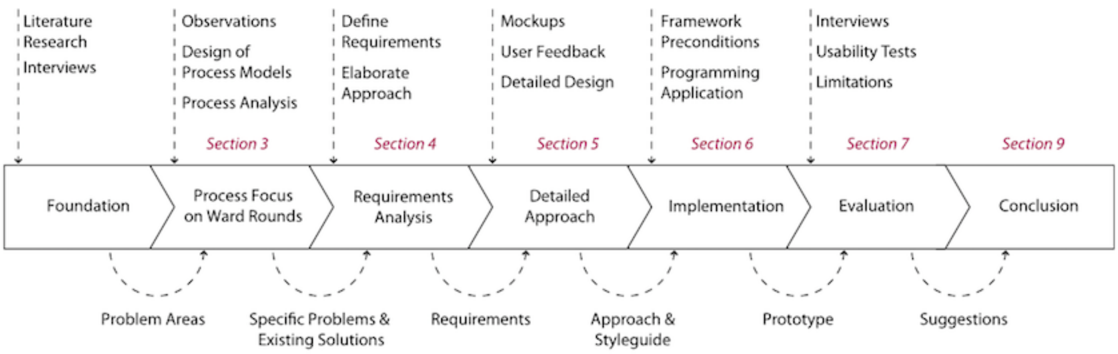


Figure 15: 7 phase of study project [PML14]

After analyzing requirement and a discussion of suitability for creating a process and mobile task support, a design of data-driven process model has been created. A high-fidelity prototype of lightweight process engine is created selecting iPad as the mobile device. After that, a presentation is given to participants about its features and participants offered to test it followed by an interview. The test has not been conducted in the real environment but rather in a simulated environment. After conducting the testing feedbacks are collected from physicians and nurses [PML14].

Based on the feedback received, physicians and nurses satisfied by MEDo project agree with it does ease overall ward round management and improve the original paper-based process [PML14]. Improving the patient data availability during the process perceived as major benefit of using mobile service.

However, there are also other issues pointed out such as concern from physicians that using mobile during ward round might distract their attention from the patient. As well as request to make user experience more intuitive and customizable, wish to Integrate the mobile service with hospital's existing information system and automate as much work as possible including the statistical analysis of collected record [PML14].

In summary, these outcomes are in line with theories mentioned in the thesis so far. Not every business process are suitable for mobilizing, there are suitable processes with high return and also not so suitable ones. User acceptance is the key, users expect seamless integration and intuitive user experiences from mobile processes meanwhile they might also distract them. Overall, in the right business process, it should simplify and speed up the process significantly.

6.4 Summary

Table 3 summarizes recommended architectures of supporting mobile devices for different services and business process types. Small scale enterprises with simple, singular service do not require to use business process model to express their business logic. It is enough to develop their service using popular client-server architectures and preferred mobile application interface.

The enterprises, which have more complex IT systems such as distributed business process execution environment collaborated by multiple partners, have two options to integrate mobile into their business processes. First one is running mobile tasks as in single mobile task handling approach. They can use MUIT architecture to simplify the creation of user interface for mobile and skip developing a mobile client for mobile

Service type	Business process type	Suggested architecture
Singular service from one enterprise	Not needed	Client-server architecture with preferred mobile application interface
Complex service from inter-enterprise collaboration	Pre-specified, repetitive business process	Single mobile task approach with backup and delegation service for robustness
	Knowledge-intensive business process	Mobile business process engine with process migration approach to distribute process execution and process migration meta-model for improving synchronization and allowing ad-hoc modification of process definitions

Table 3: Architecture Suggestion for Different Service and Business Processes

devices. For solving mobile-specific challenges and ensuring robust execution of the business process, the delegation and backup service architecture is a great option. Unfortunately, these two architectures are developed by two different research groups and currently, they are not compatible with each other. An organic combination of them can be the mobile support architecture topic of future research in business process management field.

Knowledge-intensive business processes have higher requirement for flexibility and they should be more adaptable to change. The processes that need parallel execution on both the process engine and the user's mobile require a lightweight process engine running on the user's mobile device. This is the second way of integrating mobile into process execution discussed earlier is called process migration. The MARPLE example provided an excellent reference architecture for such scenarios. Also, the MEDo project by the same working group provides proof-of-concept that such an architecture could work and could be useful in knowledge-intensive scenarios. However, user acceptance is the key, and users expect highly intuitive user experiences from mobile devices.

It is worth mentioning that the MARPLE example and the development mobile business process engines are still in the research phase. Although we have seen proof-of-concept from the MEDo project by the same working group, currently there is no commercial example of mobile business process engines in the industry [PTR10].

7 Conclusion

IT consumerization is a strong trend with benefits and risks. Companies can improve their process efficiency to a large extend by embracing mobile technologies. Because the mobile workforce has the advantage of utilizing location flexibility. However, there are risks related to information security from using personal devices, and additional stress to employees because of the constant availability and task failures caused by unreliable network connections. Embracing the mobile devices also involves a significant cost for renewing current IT infrastructure. In order to avoid the risks and utilize the benefits, companies should cautiously assess whether their business is really suitable for such a change and to what extent.

This thesis analyzed trends in mobile devices and their effects on business, identified the characteristics of mobile devices and applications and the challenges of supporting mobile devices; explained proper scenarios and benefits of adding mobile support to business and laid out the implementing mobile support roadmap for enterprises from small to big scale.

Enterprises should go through following steps during the process of realizing mobile service for their business process: learning, feasibility study, planning, integrating mobile into process, implementing mobile client and backend and improving reliability.

At learning phase, they learn what is the mobile trend and why it is important to them. What could be the effects of mobile consumerization to their business. At feasibility study phase, they should find out whether their business process is suitable for mobile support and estimate the benefit versus costs. At planning phase, enterprise should understand their options of implementing mobile service by best reusing their existing assets and formalize a strategy. They should came up with the architectural choices for their mobile client and backend server implementation. How they are going to integrate mobile devices to their business processes. After that the actual implementation based on the architectural choices and also improvements to business process execution engines in order to overcome mobile specific challenges and enable robust execution.

For small scale enterprises with simpler use cases, it is now clear that mobile web applications are the most cost-effective way to support the widest mobile audience. Following the increasing complexity of use case and business scenario they might need to consider multiple other approaches to develop mobile support in the most efficient solution.

Supporting mobile devices in process-aware information systems in big enterprises also have been studied and from pre-specified, repetitive business processes to most complex knowledge intensive process use cases now have proper technologies to adapt in order to seamless, robust execution of their business processes with mobile devices.

References

BBC15 American airlines planes grounded by ipad app error. URL http://www.bbc.com/news/technology-32513066. last checked: 12.06.2017.

BeB14 *Adoption of Mobile Business Solutions and its Impact on Organizational Stakeholders*, number 8 in BLED 2014 Proceedings, 2014. URL http://aisel.aisnet.org/bled2014/8.

BMM05 Baresi, L., Maurino, A. and Modafferi, S. *Workflow Partitioning in Mobile Information Systems*, pages 93–106. Springer US, Boston, MA, 2005. URL http://dx.doi.org/10.1007/0-387-22874-8_7.

BoP08 Botzenhardt, T. and Pousttchi, K., Analyzing the benefits of mobile enterprise solutions using the example of dispatching processes. *Mobile Business, 2008. ICMB '08. 7th International Conference on*, July 2008, pages 260–269.

CKh08 Crampton, J. and Khambhammettu, H., Delegation and satisfiability in workflow systems. *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies*, SACMAT '08, New York, NY, USA, 2008, ACM, pages 31–40, URL http://doi.acm.org/10.1145/1377836.1377842.

ChL11 Charland, A. and Leroux, B., Mobile application development: Web vs. native. volume 54, New York, NY, USA, May 2011, ACM, pages 49–53, URL http://doi.acm.org/10.1145/1941487.1941504.

CNa08 Chen, L. and Nath, R., A socio-technical perspective of mobile work. volume 7. IOS Press, 2008, pages 41–60.

IGi15 de Morais Barroca Filho, I. and Júnior, G. S. A., Development of mobile applications from existing web-based enterprise systems. volume 11, 2015, pages 162–182.

DVW05 Dumas, M., Van der Aalst, W. M. and Ter Hofstede, A. H., *Process-aware information systems: bridging people and software through process technology.* John Wiley & Sons, 2005.

FaT14 Falk, T. and Leist, S., Effects of mobile solutions for improving business processes. *ECIS 2014 Proceedings - 22nd European Conference on Information Systems*, 2014, URL `www.scopus.com`.

FWC14 Flora, H. K., Wang, X. and Chande, S. V., An investigation on the characteristics of mobile applications: A survey study. volume 6, 2014, page 21.

Gar13 Gartner says by 2016, more than 50 percent of mobile apps deployed will be hybrid, May 2015. URL `http://www.gartner.com/newsroom/id/2324917`. last checked: 12.06.2017.

GRa91 Glazer, R., Marketing in an information-intensive environment: strategic implications of knowledge as an asset. JSTOR, 1991, pages 1–19.

GoP09 Goeke, L. and Pousttchi, K., Influencing factors for the introduction of mobile-integrated business processes. *The 9th International Conference on Electronic Business (ICEB 2009), Macau, China*, 2009, pages 733–738.

HCh93 Hammer, M. and Champy, J., Reengineering the corporation. 1993. 1993.

HGK14 Hoos, E., Gröger, C., Kramer, S. and Mitschang, B., Improving business processes through mobile apps an analysis framework to identify value-added app usage scenarios. *ICEIS 2014 - Proceedings of the 16th International Conference on Enterprise Information Systems*, volume 2, 2014, pages 71–82, URL `www.scopus.com`.

HaP14 Ham, H. K. and Park, Y. B., Designing knowledge base mobile application compatibility test system for android fragmentation. volume 8, 2014, pages 303–314, URL `www.scopus.com`.

HuV12 Huy, N. P. and vanThanh, D., Evaluation of mobile app paradigms. *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, MoMM '12, New York, NY, USA, 2012, ACM, pages 25–30, URL `http://doi.acm.org/10.1145/2428955.2428968`.

IBE16 ios background execution. URL `https://developer.apple.com/library/content/documentation/iPhone/`

`Conceptual/iPhoneOSProgrammingGuide/BackgroundExecution/BackgroundExecution.html`.

JEA07 Jordan, D., Evdemon, J., Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., Ford, M., Goland, Y. et al., Web services business process execution language version 2.0. volume 11, 2007, page 5.

KoC14 Koch, H. and Curry, P., It consumerization's impact on enterprise it. *20th Americas Conference on Information Systems, AMCIS 2014*, 2014, URL `www.scopus.com`.

NBL10 Kiukkonen, N., J., B., Dousse, O., Gatica-Perez, D. and J., L., Towards rich mobile phone datasets: Lausanne data collection campaign. *Proc. ACM Int. Conf. on Pervasive Services (ICPS), Berlin.*, 2010.

KKR15 Kim, H. J., Karunaratne, S., Regenbrecht, H., Warren, I. and Wunsche, B. C., Evaluation of cross-platform development tools for patient self-reporting on mobile devices. *Conferences in Research and Practice in Information Technology (CRPIT)*, volume 164, January 2015, pages 55–61.

DRM12 Knuplesch, D., Pryss, R. and Reichert, M., Data-aware interaction in distributed and collaborative workflows: Modeling, semantics, correctness. *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, Oct 2012, pages 223–232.

KSV09 Klint, P., v. d. Storm, T. and Vinju, J., Rascal: A domain specific language for source code analysis and manipulation. *2009 Ninth IEEE International Working Conference on Source Code Analysis and Manipulation*, Sept 2009, pages 168–177.

Lau07 Laukkanen, T., Internet vs mobile banking: comparing customer value perceptions. volume 13, 2007, pages 788–797, URL `http://dx.doi.org/10.1108/14637150710834550`.

SLe09 Leroy, S., Why is it so hard to do my work? the challenge of attention residue when switching between work tasks. volume 109, 2009, pages 168 – 181, URL `//www.sciencedirect.com/science/article/pii/S0749597809000399`.

Ley10 Leymann, F., Bpel vs. bpmn 2.0: Should you care? *Business Process Modeling Notation: Second International Workshop, BPMN 2010, Potsdam, Germany, October 13-14, 2010. Proceedings*, Mendling, J., Weidlich, M. and Weske, M., editors, Berlin, Heidelberg, 2010, Springer Berlin Heidelberg, pages 8–13, URL `http://dx.doi.org/10.1007/978-3-642-16298-5_2`.

LGA12 Laurila, J. K., Gatica-Perez, D., Aad, I., J., B., Bornet, O., Do, T.-M.-T., Dousse, O., Eberle, J. and Miettinen, M., The Mobile Data Challenge: Big Data for Mobile Computing Research. *Pervasive Computing*, 2012.

RMa07 Lenz, R. and Reichert, M., {IT} support for healthcare processes – premises, challenges, perspectives. volume 61, 2007, pages 39 – 58, URL `http://www.sciencedirect.com/science/article/pii/S0169023X06000784`. Business Process Management Where business processes and web services meet.

LWR14 Lanz, A., Weber, B. and Reichert, M., Time patterns for process-aware information systems. volume 19, 2014, pages 113–141, URL `http://dx.doi.org/10.1007/s00766-012-0162-3`.

LXH16 Liu, X., Xu, M., Huang, G., Teng, T., Zheng, Z. and Mei, H., MUIT: A middleware for adaptive mobile web-based user interfaces in WS-BPEL. volume abs/1602.09125, 2016, URL `http://arxiv.org/abs/1602.09125`.

MFB09 Malik, M., Frosch-Wilke, D., Beck, S., Hartmann, C., Sturm, T. and Wieben, T., Mobile applications in the german health insurance system to improve the market position. *HEALTHINF 2009 - Proceedings of the 2nd International Conference on Health Informatics*, 2009, pages 117–122, URL `www.scopus.com`.

MHL10 Maurer, M.-E., Hausen, D., De Luca, A. and Hussmann, H., Mobile or desktop websites?: Website usage on multitouch devices. *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, New York, NY, USA, 2010, ACM, pages 739–742, URL `http://doi.acm.org/10.1145/1868914.1869018`.

DDF08 Martin, D., Wutke, D. and Leymann, F., A novel approach to decentralized workflow enactment. *2008 12th International IEEE Enterprise Distributed Object Computing Conference*, Sept 2008, pages 127–136.

FJA12 Nayebi, F., Desharnais, J. M. and Abran, A., The state of the art of mobile application usability evaluation. *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, April 2012, pages 1–4.

OKM13 Ortbach, K., Köffer, S., Müller, C. P. F. and Niehaves, B., How it consumerization affects the stress level at work: A public sector case study. *PACIS*, 2013, page 231.

PoH09 Pousttchi, K. and Habermann, K., Exploring the organizational effects of mobile business process reengineering. *Mobile Business, 2009. ICMB 2009. Eighth International Conference on.* IEEE, 2009, pages 353–358.

PLR13 Pryss, R., Langer, D., Reichert, M. and Hallerbach, A. *Mobile Task Management for Medical Ward Rounds – The MEDo Approach*, pages 43–54. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. URL `http://dx.doi.org/10.1007/978-3-642-36285-9_6`.

PML14 Pryss, R., Mundbrod, N., Langer, D. and Reichert, M., Supporting medical ward rounds through mobile task and process management. 2014, pages 1–40, URL `www.scopus.com`. Article in Press.

PMR14 Pryss, R., Musiol, S. and Reichert, M., Integrating mobile tasks with business processes: A self-healing approach. In *Handbook of Research on Architectural Trends in Service-Driven Computing*, August 2014, pages 103–135, URL `http://dbis.eprints.uni-ulm.de/1095/`.

PMR16 Pryss, R., Musiol, S. and Reichert, M., Extending business processes with mobile task support: A self-healing solution architecture. IGI Global, 2016, page 273.

PRB15 Pryss, R., Reichert, M., Bachmeier, A. and Albach, J., Bpm to go: Supporting business processes in a mobile and sensing world. In *BPM Everywhere*, 2015, pages 167–182, URL `http://dbis.eprints.uni-ulm.de/1153/`.

PTK11 Pryss, R., Tiedeken, J., Kreher, U. and Reichert, M., *Towards flexible process support on mobile devices*, volume 72 LNBIP of *Lecture Notes in Business Information Processing*. 2011.

PTR10 Pryss, R., Tiedeken, J. and Reichert, M., Managing processes on mobile devices: The marple approach. *CAiSE'10 Demos*, June 2010, URL `http://dbis.eprints.uni-ulm.de/663/`.

QtA15 Qt application development, May 2015. URL `http://www.qt.io/application-development/`. last checked: 12.06.2017.

RBa07 Reichert, M. and Bauer, T., Supporting ad-hoc changes in distributed workflow management systems. *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS: OTM Confederated International Conferences CoopIS, DOA, ODBASE, GADA, and IS 2007, Vilamoura, Portugal, November 25-30, 2007, Proceedings, Part I*, Meersman, R. and Tari, Z., editors, Berlin, Heidelberg, 2007, Springer Berlin Heidelberg, pages 150–168, URL `http://dx.doi.org/10.1007/978-3-540-76848-7_11`.

RGG13 Rivera, J., Goasduff, L. and Gartner, G., Gartner says worldwide pc, tablet and mobile phone combined shipments to reach 2.4 billion units in 2013. volume 116, pages 265–731.

RHM04 Ran, L., Helal, S. and Moore, S., Drishti: an integrated indoor/outdoor blind navigation system and service. *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*. IEEE, 2004, pages 23–30.

RWe12 Reichert, M. and Weber, B., *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. URL `http://dx.doi.org/10.1007/978-3-642-30409-5_3`.

SBB13 Sammer, T., Brechbühl, H. and Back, A., The new enterprise mobility: Seizing the opportunities and challenges in corporate mobile it. *19th Americas Conference on Information Systems, AMCIS 2013 - Hyperconnected World: Anything, Anywhere, Anytime*, volume 5, 2013, pages 3452–3459, URL `www.scopus.com`.

SRP13 Schobel, J., Ruf-Leuschner, M., Pryss, R., Reichert, M., Schickler, M., Schauer, M., Weierstall, R., Isele, D., Nandi, C. and Elbert, T., A generic questionnaire framework supporting psychological studies with smartphone technologies. *XIII Congress of European Society of Traumatic Stress Studies (ESTSS) Conference*, June 2013, pages 69–69, URL `http://dbis.eprints.uni-ulm.de/962/`.

SSP13 Schobel, J., Schickler, M., Pryss, R., Nienhaus, H. and Reichert, M., Using vital sensors in mobile healthcare business applications: Challenges, examples, lessons learned. *9th Int'l Conference on Web Information Systems and Technologies (WEBIST 2013), Special Session on Business Apps*, May 2013, pages 509–518, URL `http://dbis.eprints.uni-ulm.de/918/`.

SSP14 Schobel, J., Schickler, M., Pryss, R., Maier, F. and Reichert, M., Towards process-driven mobile data collection applications: Requirements, challenges, lessons learned. *10th Int'l Conference on Web Information Systems and Technologies (WEBIST 2014), Special Session on Business Apps*, April 2014, pages 371–382, URL `http://dbis.eprints.uni-ulm.de/1036/`.

SGe88 Stalk, G., Time–the next source of competitive advantage. Harvard Business Review, 1988.

APM03 van der Aalst, W. M. P., ter Hofstede, A. H. M. and Weske, M., Business process management: A survey. *Business Process Management: International Conference, BPM 2003 Eindhoven, The Netherlands, June 26–27, 2003 Proceedings*, van der Aalst, W. M. P. and Weske, M., editors, Berlin, Heidelberg, 2003, Springer Berlin Heidelberg, pages 1–12, URL `http://dx.doi.org/10.1007/3-540-44895-0_1`.

VHa10 Völzer, H., An overview of bpmn 2.0 and its potential use. *Business Process Modeling Notation: Second International Workshop, BPMN 2010, Potsdam, Germany, October 13-14, 2010. Proceedings*. Springer, 2010, pages 14–15.

WeL12 Weiß, F. and Leimeister, J. M., Consumerization: It innovations from the consumer market as a challenge for corporate it. volume 4, 2012, pages 363–366, URL `www.scopus.com`.

Xam15 Xamarin cross platform, May 2015. URL http://xamarin.com/. last checked: 12.06.2017.

You14 Youtube (2014) youtube press statistics, May 2015. URL http://www.youtube.com/yt/press/statistics.html. last checked: 10.08.2015.

ZKK10 Zaplata, S., Hamann, K., Kottke, K. and Lamersdorf, W., Flexible execution of distributed business processes based on process instance migration. volume 1, 2010, pages 3–16, URL http://search.proquest.com/docview/1690666786?accountid=11365.

ZBr99 Zimmerman, J. B., Mobile computing: Characteristics, business benefits, and the mobile framework. volume 10, 1999, page 12.

ZKo13 Zamula, D. and Kolchin, M., Mnemojno - design and deployment of a semantic web service and a mobile application. *14th Conference of Open Innovation Association FRUCT*, Nov 2013, pages 171–176.

ZKM10 Zaplata, S., Kottke, K., Meiners, M. and Lamersdorf, W., Towards runtime migration of ws-bpel processes. *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops: International Workshops, ICSOC/ServiceWave 2009, Stockholm, Sweden, November 23-27, 2009, Revised Selected Papers*, Dan, A., Gittler, F. and Toumani, F., editors, Berlin, Heidelberg, 2010, Springer Berlin Heidelberg, pages 477–487, URL http://dx.doi.org/10.1007/978-3-642-16132-2_45.